Explaining Credit Card Fraud Decisions in ML: An Analysis of XAI Methods

Ciaran Finnegan - D21124026

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Science)

April 2024

Declaration

I certify that this dissertation, which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: 

Date: 12th April 2024

Abstract

The Covid pandemic accelerated an already rapidly evolving trend towards a cashless society and simultaneously led to an even greater demand for more effective online fraud protection measures. The Financial Services industry needs to maintain its own momentum in fraud prevention by screening online transactions with increasingly sophisticated Machine Learning algorithms. At the same time, these service providers do not have a free hand in the implementation of fraud prevention applications, as regulators and the public alike still demand accountability from industry AI. This is a theme that looms even larger as the presence of GenAI becomes ever more ubiquitous in the modern world, and a certain suspicion around the perceived reliance in AI grows as the first quarter of the 21st century ends.

Credit card fraud detection through ML technology has been a commonplace and active area of research for nearly two decades. This is heavily driven by the persistent and growing threat from *'bad actors'* in this domain and the billions of Euros lost each year by individuals and Financial Institutions. Access to better sources of data and more sophisticated models continues to improve card fraud detection rates. However, the challenges from criminals continue to evolve, and financial institutions increasingly rely on concepts such as Artificial Neural Networks (ANN) to increase the speed and accuracy of credit card fraud detection. Even a cursory review of academic research in the area of credit card fraud prevention will shine a light on the dilemma companies face in this particular domain of crime prevention. It is not enough to stop a suspected instance of credit card fraud; a company must be able to explain *why*. The workings of ANN models are not readily apparent, and trusting in the *'black box'* alone will not suffice. Also, increasingly, it is not just a case of justifying a decision on fraud to an

individual customer; as industry and government regulators will likewise demand that such decision-making processes be transparent and comprehensible.

Companies that deliver applications to the financial crime prevention business sector have been cognisant in very recent years of the need to add explainability into their product suite. Vendors (such as IBM, Actimize, SymphonyAI, etc.) will boast of advances in detection rates in areas such as credit card fraud, but have also started to supplement these offerings with built-in explanation data for fraud investigators. Driven by regulatory requirements, company auditors will demand that Financial Institutions can demonstrate a developing process to prevent credit card crime, but can also justify decisions as to why a client's card transaction has been delayed/rejected (or why not). However, the development of commercial products for embedded explanations in fraud detection tools is still in a relatively nascent stage. Data scientists working in this commercial sphere are aware of the various Explainable Artificial Intelligence (XAI) techniques that can be applied in the domain of credit card fraud, but there is still relatively little published research on the comparative benefits of such approaches in this context. Furthermore, a significant volume of research focuses on the human interpretation of XAI output, regardless of whether the subject is fraud detection or health care prediction. Human surveys are costly to implement and can be susceptible to the bias and/or lack of domain knowledge by the participant. The focus of this paper is to examine an automated and statistical comparison of established XAI methods for credit card fraud and to assess whether there is a quantitative difference between them in terms of general performance. Such an analysis could provide guidance for future product roadmaps in the commercial online fraud prevention space.

Keywords: Explainable Artificial Intelligence, Credit Card Fraud Detection, Interpretability, XAI Statistical Comparison

Acknowledgments

I would like to express my sincere thanks to Dr. Bujar Raufi and Dr. Luco Longo for their academic guidance with this dissertation.
In addition, I wish to acknowledge my colleagues in SymphonyAI; Dr Rory Duthie, Eddie Baggot, Kieran MacKenna and Dan Branley for their suggestions and provocations.

For Delphine; for her patience.

Dataset: Sourced and used with permission from 2015 product research conducted by Norkom Technologies on emerging fraud detection techniques.

Contents

List of Figures

List of Tables

List of Acronyms

XAI	Explainable Artificial Intelligence
ANN	Artificial Neural Network
SHAP	SHapley Additive exPlanations
LIME	Local Interpretable Model-agnostic Explanations
DiCE	Diverse Counterfactual Explanations

Chapter 1

Introduction

1.1 Background

Credit card fraud costs the Financial Services industry billions of Euros in losses each year (Nesvijevskaia, Ouillade, Guilmin, & Zucker, 2021). The need for ever more sophisticated Machine Learning techniques to tackle this problem has been well established by academic observers such as (Dal Pozzolo et al., 2014) and (P. Sharma & Priyanka, 2020). The research of (A. Sharma & Bathla, 2020) and (Batageri & Kumar, 2021) is an example of work in this field to improve fraud detection rates through increasingly sophisticated neural network algorithms. However, many researchers highlight the parallel challenge that these *'black box'* models need to be held accountable for the individual fraud classifications they make (T.Y.Wu & Y.T.Wang, 2021).

(Ignatiev, 2020) focuses on the need for Explainable Artificial Intelligence (XAI) to be *trustable*, while (Carvalho, Pereira, & Cardoso, 2019) are more emphatic about the legal demands of the European Union that all automated decision-making about citizens be *transparent*.

This dissertation will focus on ML-driven software used by the Financial Services industry and whether an objective rating can be given to different XAI methods in terms of explaining the reason for a given credit card fraud classification. To further narrow the field of interest, the paper will propose a series of metrics to rate the

performance of four state-of-the-art XAI methods; SHAP, LIME, ANCHORS, and DiCE, on an industry credit card fraud dataset, as applied to the classification of individual credit card transactions. (These XAI approaches are described, with supporting references, in Section 2.2). Companies operating in financial crime software, such as SymphonyAI and Actimize, already sell ML-based software to detect credit card fraud, but generally rely on only one explainer technique, usually SHAP values.

Specifically, the scope of experiments in this thesis is on explanations for individual (*'local'*) transactions and only considers interpretability techniques that are *agnostic* about the type of the detection model.

1.2 Research Project/Problem

1.2.1 Research Question

"To what extent can we quantify the quality of contemporary machine learning interpretability techniques, providing local, model-agnostic, and post-hoc explanations, in the classification of credit card fraud transactions by a 'black box' Neural Network ML model?"

The question will focus on a quantitative comparison of explanations produced by different XAI techniques on specific (local) NN model predictions.

1.2.2 Research Problem

The research problem can be described as the means to produce an objective assessment of state-of-the-art ML explainers, as applied to credit card fraud detection. The purpose is to compare a set of common XAI techniques and to find insights into the relative strengths of each one. The focus of the experiment is on the application of SHAP, LIME, ANCHORS, and DiCE interpretability methods to a neural network model trained on a commercial dataset containing credit card transactions, which are labelled *'fraud'* or *'non-fraud'*.

1.3 Research Objectives

Metrics for all four explainer techniques will be collated based on the methodology described in Section 1.4 below. This output will be subjected to a statistical test for significance. Is one explainer better than another and, if so, how great is that difference?

The objective of the experiments underpinning this research paper is a very premeditated intention to avoid any elements of human assessment of the XAI techniques. Taking the lead from a study of XAI techniques applied to eye tracking experiments in research by (Martínez, Nadj, Langner, Toreini, & Maedche, 2023) this research will aim to demonstrate that it is possible to produce a purely statistical analysis of explainer performance for individual credit card fraud predictions.

1.4 Research Methodologies

The experiments proposed in this article are based on a similar study on measuring interpretability methods on healthcare datasets that classified mortality predictions (ElShawi, Sherif, Al-Mallah, & Sakr, 2020). This study defined a set of generic and custom metrics and then *scored* the output of a series of XAI techniques against these metrics.

As listed in Section 1.1, the experiments in this research paper will score SHAP, LIME, ANCHORS, and DiCE methods on an industry credit card fraud dataset. The metrics for the experiments are defined in Section 3.2.4 of this paper and are inspired by the aforementioned research (ElShawi et al., 2020) and the XAI benchmarks produced in a paper by (Jacob et al., 2021).

A key assumption is that this research approach will translate into the domain of credit card fraud detection.

1.5 Scope and Limitations

Focus on Four Specific XAI Techniques

Experiments are being specifically limited to four post-hoc and local interpretability frameworks. Thus, the research only looks at explanations extracted from a trained model for individual credit card fraud predictions. This is done to focus on the 'business' objectives of this paper, as elaborated in the Abstract, and in particular to build on related research papers by (Ribeiro, Singh, & Guestrin, 2016) and (Guidotti et al., 2019).

Only Local Explanations Measured

Only local explanations of specific credit card transactions are being considered; global explainability of the overall model is not in scope. The potential use case for the results of this research is to improve the information with which fraud investigators are presented on specific transaction assessments. The choice of explainers for this paper has also been influenced by the graphical presentation of the most common XAI techniques for the local assessment of NN models provided in the research by (Ras, Xie, Gerven, & Doran, 2022).

Automated Experiments

Deliberately, there is no human assessment of the explanations, as this will be a purely programmatic and arithmetic exercise. This is a conscious research decision to implement a purely statistical analysis. The rationale is to generate a lower cost assessment framework for XAI output but to also avoid situations such as those described by (Chromik, .Eiband, Buchner, Krüger, & Butz, 2021) whereby users surveyed fall foul of that the authors describe as *"...the illusion of Explanatory Depth in Explainable AI..."*.

Potential Hardware Limitations

If the use of extensive GPU processing was required for certain explainers, then this may have warranted an adjustment in experiment scope. However, this did not prove to be an issue with the final set of experiments.

1.6 Document Outline

This dissertation begins with a critical review of relevant recent research conducted in the area of Explainable AI (XAI), with a particular focus on the evaluation of common explainer methods applied to model predictions. The research topic of this paper is strongly focused on techniques for local model-agnostic interpretations of classification model results, which has informed the literature reviewed and evaluated in the following chapter (Chapter 2).

Chapter Three describes the experimental approach to build an evaluation matrix for the four chosen explainer techniques; SHAP, LIME, ANCHORS, and DiCE (Counterfactuals). Five sets of metrics are generated based on the explanation outputs of each technique for predictions on a binary credit card fraud classification problem.

The methodologies described in Chapter Three deliver the XAI metrics output, which are then assessed in Chapter Four. A significance test is conducted to evaluate the relative effectiveness of each technique. This analysis is supplemented by a further statistical calculation as to how these approaches can be ranked in any order of merit.

The paper concludes in Chapter 5 with a consolidation of these results and a possible future application of the analysis, along with other recommended avenues of investigation.

Chapter 2

Review of existing literature

2.1 Key Themes in Current Research

2.1.1 How to Measure the Effectiveness of an Explanation? No Obvious Consensus

The literature review for this dissertation began with assessments of how the detection of credit card fraud using machine learning models is being refined with ever more sophisticated neural network models (P. Sharma & Priyanka, 2020). However, in their research experiments with the LIME algorithm, (Ribeiro et al., 2016) describe how users can have a trust issue with NN ML models because they are effectively *'black-boxes'* from which it is very difficult to interpret why a given classification has been derived. This is a theme echoed in the introduction of many research papers, such as (ElShawi et al., 2020), (Honegger, 2018), and (Sinanc, Demirezen, & Sağıroğlu, 2021). Despite these acknowledgements, in this research domain there appears to be no cast iron process to establish XAI *trustworthiness*. Although universal frameworks to interpret model predictions have been proposed (Lundberg & Lee, 2017) there is still no unanimity seen in research to date on what constitutes an objectively *'good'* explanation of a prediction. The gap remains; How exactly does a researcher measure and display *'explainability'* in Explainable Artificial Intelligence (XAI) research?

To further emphasise this gap in contemporary research, (Adadi & Berrada, 2018) claimed that *"Technically, there is no standard and generally accepted definition of explainable AI"* (p. 141). More specifically, in their review of XAI research papers, (Vilone & Longo, 2021b) state that *"There is not a consensus among scholars on what an explanation exactly is and which are the salient properties that must be considered to make it understandable for every end user'* (p.651) Therefore, as stated above, there is no well-established output framework to explain credit card fraud classification through *'black-box'* models (Vilone & Longo, 2021a).

This paper proposes to build on some of the objective research on scoring predictions generated by four established interpretability methods.

2.1.2 Human Assessment vs Automated Benchmarks

Research by (Jacob et al., 2021) makes the point about assessing XAI output that *"...while a user study may be the best way to evaluate the usefulness of explanations, it is not always available and may come at a high cost."*. It is also desirable for humans participating in XAI surveys to have some degree of domain knowledge, but fraud detection explainer experiments by (Jesus et al., 2021) showed that this can still be subject to user bias.

Examples of XAI research where the reliance on human assessment of explanations is less commonplace can be seem in the domain of healthcare, through research by (Marcilio & Eler, 2020) and (Lakkaraju, Bach, & Leskovec, 2016). Those experiments produce clear objective recommendations in line with the work of (ElShawi et al., 2020). Research into explanations for ML fraud classification often follows a more subjective survey style of experimentation involving the augmentation of human-based processes with model explainer outputs. On occasion, human bias can have an adverse impact on the reliability of the interpretation of the model explanations generated by ML (Kaur et al., 2020). This dissertation will follow the steps of previous research that use only non-human programmatic experiments with quantifiable metrics (Darias, Caro-Martínez, Díaz-Agudo, & Recio-Garcia, 2022) and

tests of statistical significance (Evans, Xue, & Zhang, 2019).
The methodology for the experiments in this paper is based heavily on the 2020 XAI research in healthcare from Elshawi et al., but also takes inspiration from the anomaly detection framework for explainability created by (Nguyen et al., 2023) and the XAI time series results produced by (Schlegel, Arnout, El-Assady, Oelke, & Keim, 2019), both using SHAP and LIME explainer outputs. A further bespoke framework can be seen in the research by (Moreira et al., 2020) using local explanations also generated with SHAP, LIME and Counterfactual techniques.
The use of statistical significance tests, which form the basis of the evaluations of the experiments in Section 4.3 of this document, follows the ML evaluation processes described by (Hanafy & Ming, 2022).
The metrics described in Section 3.2.4 of this article are also influenced by the aforementioned healthcare research, but have been augmented by the concepts of 'robustness' expounded by (Alvarez-Melis & Jaakkola, 2018).

2.1.3 Neural Networks and XAI

To reflect that credit card fraud detection models are based on increasingly more sophisticated NN algorithms (Ajitha, Sneha, Makesh, & Jaspin, 2023) (Aurna, Hossain, Taenaka, & Kadobayashi, 2023), the experiments in this paper will involve local *post-hoc* explanations generated on a trained NN model. For this paper, a significant body of research material on XAI approaches using Deep Learning (DL) techniques was evaluated. This helped direct the experiments described in Section 3.2 on the use of procedures such as Deep SHAP (Nascita et al., 2021) (Sullivan & Longo, 2023), LIME for Deep Neural Network (DNN) experiments (Ras et al., 2022), and the use of counterfactuals in CNN model explainers (Vouros, 2022).

2.1.4 Computational Efficiency

Also of note is the observation from (Psychoula et al., 2021) that the runtime implications of XAI output on real-time systems, fraud or otherwise, have had

relatively little research focus to date. Early prototyping in this dissertation effort will attempt to capture and address such issues as quickly as possible.

2.1.5 Presenting XAI Data

(Guidotti et al., 2019) conducted comparative experiments into local interpretability frameworks, but note in their conclusions that there is still relatively little research on building more aesthetically attractive visualisations of such explanations. This will not be a focus area of this dissertation.

2.2 State of the Art Approaches for Local Interpretability

This section of the document describes research conducted on local interpretability techniques that formed the basis of the experiments in this dissertation research.

2.2.1 SHAP

SHAP stands for **SH**apley **A**dditive ex**P**lanations (Lundberg & Lee, 2017) and can be described as a unified framework for interpreting predictions. It provides a toolkit that is computationally efficient in calculating the 'Shapley' values. SHAP is a method derived from cooperative game theory, and SHAP values are used extensively to present an understanding of how features in a dataset are related to the model prediction output. It is a *'black box'* explainability technique that can be applied to most algorithms without knowing the exact model.
The focus of this dissertation research is on local interpretations, so we will be using SHAP to understand how the NN model made a fraud classification for a single transaction instance. (SHAP values can also be used for global interpretations of a given model).

2.2.2 LIME

LIME stands for **L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations (Ribeiro et al., 2016) and is also a popular choice for interpreting decisions made by black box models. The core concept of LIME is that it aims at understanding the features that influence the prediction of a given black-box model around a single instance of interest. LIME approximates these predictions by training local surrogate models to explain individual predictions.

2.2.3 ANCHOR

ANCHORS was also developed by Marco Ribeiro (Ribeiro, Singh, & Guestrin, 2018) and is, again, a model-agnostic explanation approach based on if-then rules that are called *'anchors'*. These *'anchors'* are a set of feature conditions that act as high precision explainers created using reinforcement learning methods. This process is more computationally demanding than SHAP but is considered to have a better generalisability than LIME.
There is a perception that Anchors provide a set of rules that are more easily understood by end users, although in this dissertation the analysis will be solely on the comparison of quantitative metrics. This will require the conversion of Anchor data into a numerical format for comparative statistical analysis.

2.2.4 DiCE

DiCE (Diverse Counterfactual Explanations) (Mothilal & Tan, 2020) is an XAI method developed to provide information on the decisions of the machine learning model by generating counterfactual explanations. In essence, a counterfactual explanation describes a minimal set of changes required to alter the model's prediction for a particular instance. For example, in a loan approval scenario, if a model classification result declined an applicant, DICE could elucidate that increasing the annual income by a specific amount or improving the credit score by a few points would have led to an approval. This approach not only aids in

understanding the model's behaviour but also provides actionable feedback to the end-users.

The strength of DiCE lies in its ability to produce diverse counterfactuals that span the different dimensions of the feature space, allowing stakeholders to obtain a holistic view of the model's decision-making process (Nri, Jenkins, Paul, & Caruana, 2019).

Again, for the purposes of this paper, experiments with DiCE explanations will be converted into numerical input for statistical analysis.

Chapter 3

Experiment design and methodology

3.1 Research Hypothesis for this Paper

Null Hypothesis:

It is not possible to quantify and distinguish the statistically best interpretation framework to explain the reason for a specific (local) credit card fraud classification result using the following state-of-the-art techniques; SHAP, LIME, ANCHORS, and DICE.

Alternate Hypothesis:

IF a Neural Network algorithm is trained on a credit card transaction dataset for ML fraud detection, and SHAP, LIME, ANCHORS, and DICE interpretability frameworks are applied to individual model results.

THEN a test for significance can be applied to the scores of each interpretability framework, against a predefined set of quality metrics, to rank each explainer technique and to demonstrate statistically which is best for explaining local credit card fraud classification results, based on the individual custom metrics.

Section 3.2.4 of this paper provides the list of evaluation metrics that will be used to measure the performance of each explainer technique in the experiments for this paper.

A Friedman Test will be applied across the four XAI techniques using lists of custom metric scores to rank the interpretability outputs for SHAP, LIME, ANCHORS, and DICE. A P-value output of this test of less than 0.05 will be considered sufficient evidence against the Null Hypothesis in favour of the Alternate (or vice versa).
The P-value alone is not sufficient for this investigation, as it will be necessary to determine the degree of performance separation between the interpretability frameworks. A Wilcoxon signed rank test will be applied pairwise on the interpretability techniques to measure the scale of difference, if any, in performance between each explainer method and between individual custom score results.

3.2 Design and Implementation

3.2.1 Research Objectives and Experimental Activities

The research aim is to rank four selected interpretability frameworks (LIME, SHAP, Anchors, and DiCE), using predefined, custom-built comparison metrics, against the output of a Neural Network (NN) credit card fraud detection model and determine which one, if any, demonstrates the best overall performance.
The study will run an iteration of the eight following research steps to compile a table of metric results for each explainer method.

These steps will build a statistical comparative analysis of the performance by each technique;

1. Train, test, and evaluate a credit card fraud detection model, built with a neural network model. This is a highly performant binary classification predictive model around which each individual explainer method will generate the explanations to be analysed.

2. Generate explanation(s) for each method based on either a single instance or a very small subset, taken from the test data. Use this output to create a visual display of the explanation to validate that the explainer is generating a meaningful end product.

3. Refine the credit card fraud model building process, if necessary, to improve the quality of the explanations without compromising model performance.

4. Break out the test data into equal blocks of feature instances, with associated fraud labels, and generate explanations for the instances in each block for each separate XAI method. The experiment metrics defined for use in this thesis require a fully numerical input, so the output of some of the XAI methods will be converted as appropriate to numerical values. The reason for working with blocks of test data is that some of the XAI methods are computationally very heavy, and processing the entire test dataset all at once was impractical.

5. Submit the XAI output data to a separate Python function to generate a value from each experiment metric (Identity, Stability, Seperability, Similarity, and Computation Efficiency - see 3.2.4 for further elaboration).

6. Review the metric scores to determine whether there was any distortion during the conversion of the XAI method output to numerical values. Correct as appropriate and regenerate the XAI metric scores.

7. Take the metric scores of each block of data for each individual XAI technique and use these values as input for a statistical significance comparison.

8. Conduct a comparative statistical analysis of the XAI metric score for each XAI method and determine if any significant performance difference can be proven. This analysis will dictate the key results and observations of this thesis.

The research focus is on explanations for fraud classification of individual transaction records, hence these experiments only consider local, post-hoc results, and not an analysis of how the overall model can explain result outcomes.

3.2.2 Data Source and Ethical Considerations

The dataset for this study was obtained from my employer, SymphonyAI, but is related to a product development cycle that ran from 2014 to 2018 by a subsidiary

company (Norkom Technologies). The data was synthesised in 2013 from various sources based on US credit card transactions and contains 25,128 rows, each representing a credit card purchase. In this set of records, 15% of the entries have been labelled as *'fraud'* by an historical analysis of the types of transactions that were later reported as fraudulent. The data was used for product testing and demonstration purposes, but that particular product line was discontinued in 2019 and permission has been granted to access this, now redundant, dataset.
The synthetic data generation process did not replicate any Personally Identifying Information (PII) from external sources.
The 2013 data generation process brought in a significant amount of POS information, along with certain ETL attributes for use within the Norkom fraud application, resulting in a dataset of 380 columns. With such high levels of dimensionality, it is necessary to prune the feature set and focus on the most important data columns. This is important not only for the accuracy of the Neural Network model, but also because XAI explainer techniques do not perform well on datasets with a high number of features.
The source data has an extremely low number of missing values, which are quickly removed at the start of the experiments, and is free of any corruption in the data elements. The *'fraud'* label is a simple *'0'* or *'1'* binary value, *'1'* being used to represent that this transaction record was deemed fraudulent. Therefore, the model-building exercise is a standard predictive classification problem.

3.2.3 Experiment Design: Generating XAI Metrics

Building a NN Model + Extraction of Explanations (x4)

The initial experiment steps will be to re-engineer the data prior to model creation. Credit Card fraud datasets often suffer from a severe imbalance of the target class (Priscilla & Prabha, 2020). However, the fraudulent records in this research dataset comprise 15% of the entire data. While this is considerably more balanced than typical credit card fraud datasets, we will down-sample the non-fraud records to

create an even classification split. To simplify the process and avoid adding any new synthetic data, a section of non-fraud records will be removed so that the remaining data set is 7K rows in size with a 50/50 breakdown of fraud vs. non-fraud. (Ribeiro et al., 2016) note that high-dimensional data can complicate the interpretability process and it is generally desirable to focus on the key features for local explainer outputs.

All 7K records will be used for model training, testing, and refinement. Following a standard Machine Learning workflow process, the dataset is split into a 80:20 ratio for training:testing.

Using Kubeflow Notebooks (described below), a basic classifier model is created and used to identify and remove unnecessary highly correlated features. As discussed above, dimensionality reduction is also an important consideration in improving the XAI output. Thus, a *'sub-experiment'* will first take place to identify the 40 most important features in this classification problem. The results of this preparatory exercise will be used to limit the number of features processed in the *'main'* set of experiments.

Taking comparative NN fraud detection experiments from (Sinanc et al., 2021) and (Anowar & Sadaoui, 2020), an ideal target performance threshold of close to >= **0.85** and >= **0.85** will apply for **Accuracy** and **F1** values respectively.

Although it is very commonplace for credit card fraud datasets to be highly imbalanced, with often very few instances of fraudulent behaviour, this is not the case in the dataset used for the experiments in this thesis. Therefore, the general performance of the model as measured by the overall **accuracy** in predicting both *fraud* and *non-fraud* outcomes is a meaningful statistic.

The **F1** score is the harmonic mean of precision and recall. Precision is the number of true positives (correctly identified positive cases) divided by the total number of positive predictions (true positives plus false positives). Recall, also known as *sensitivity* or the true positive rate, is the number of true positives divided by the total number of actual positives (true positives plus false negatives). In the detection of credit card fraud, both precision and recall are important. High precision means

that when the model predicts fraud, it is likely to be correct, minimising inconvenience for customers. High recall means that the model is good at catching fraudulent transactions, which is critical for financial security. In fraud detection, the *positive* class (fraudulent transactions) is generally of greater importance and interest. The F1 score specifically focuses on the model's performance in predicting this class. Although these particular XAI experiments have the luxury of working with a well-balanced dataset, the objective of accurate fraud detection is a strong imperative in the Financial Services sector and model performance here should reflect that expectation.

The above evaluation criteria will ensure that a performant NN model has been created before the measurements of the results from the experiments on the separate interpretability frameworks.

Model building will follow a standard iteration of feature engineering, feature selection, and NN (Keras) hyperparameter tuning to ensure the above target criteria are achieved.

The importance of model performance is also critical to the quality of the output of the XAI method. If the model is not performant (guilty of underfitting or overfitting the training data, for example), then the meaningfulness of the resultant explanations will be compromised.

The evaluation statistics of the Neural Network model used in the experiments are presented in Section 4.1 of this thesis report.

Assessing the Explanations from Each XAI Method

A single predictive model for credit card fraud will be generated once in a single dedicated Python Notebook, and stored for use in experiments to generate explainer output for these XAI methods;

1. SHAP
2. LIME
3. ANCHORS

4. DiCE (counterfactuals)

The 1,400 records in the test data block will subsequently be subdivided into 20 batches for use in the research experiments to generate a table of numerical outputs against the following metrics (elaborated in Section 3.2.4 of this thesis);

1. Identity
2. Stability
3. Separability
4. Similarity
5. Computational Efficiency

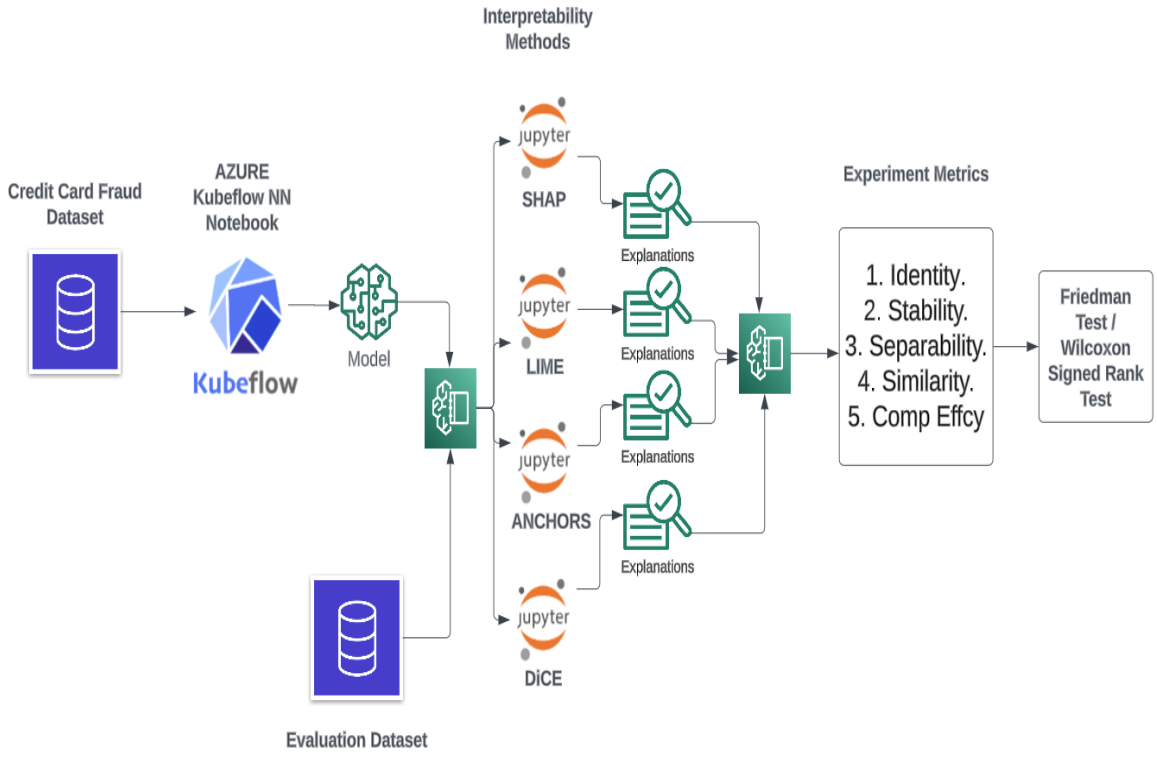


Figure 3.1: Overview of experiment design

Figure 3.1 shows the diagrammatic view of the experiment design to compare explainability methods.

The experiments will be created and executed within a Kubeflow Studio integrated development environment (IDE). Kubeflow offers a Jupyter Notebook-style interface and the experiments will be written using Python 3.7. The resources assigned to each notebook kernel will be identical, particularly so that the *'Computational Efficiency'* metric can be compared accurately across all explainer techniques.
The SHAP, LIME, ANCHORS, and DiCE explainability techniques are used to generate the explanations from a Neural Network model, built with a Python Keras library. However, the generation of a number of the XAI explanations is a processor intensive exercise, and even attempting to generate both explanations and metrics on the single 1.4K data block was found to be impractical and prone to system timeouts. As each of the 20 equal-sized test data chunks is scored against the XAI metrics, the results are written to a separate external file for each SHAP, LIME, ANCHORS, and DiCE technique. The 20 results for each metric, against each XAI explainer, are collated to provide the final *scores* for each explainer/metric combination. The table in Section 4.2 illustrates the final numerical output of the experiment phase.

3.2.4 Experiment Design: Evaluation of XAI metrics and Statistical Analysis

The explainability metrics proposed below extend the framework comparison research conducted by (ElShawi et al., 2020), but transfer the domain from healthcare analysis to credit card fraud detection. (ElShawi et al., 2020) was in turn influenced by papers from (Honegger, 2018) and (Guidotti et al., 2019).
The purpose of the research is to gather knowledge from the numerical results of the experiments and determine whether the explanation frameworks can be clearly ranked in terms of overall performance by the applied metrics. As described above, this approach follows some of the concepts in measuring similarity performance for explainability techniques elaborated by (ElShawi et al., 2020) and emulated in this dissertation.
For this paper, the metrics in the research referenced above have been adapted and

extended to measure the following XAI characteristics;

1. **Identity**. A measure of how many identical instances have identical explanations. For every two instances in the testing data, if the distance between features is equal to zero, then the distance between the explanations should be equal to zero.

2. **Stability**. Instances belonging to the same class have comparable explanations. K-means clustering is applied to the explanations for each instance in the test data. Measure the number of explanations in both clusters (fraud/non-fraud) that match the predicted class.

3. **Separability**. Dissimilar instances must have dissimilar explanations. Take a subset of test data and determine for each individual instance the number of duplicate explanations in the entire subset, if any.

4. **Similarity**. This metric captures the assumption that the more similar the instances to be explained, the closer their explanation should be (and vice versa). Cluster test data instances into Fraud/non-Fraud clusters. Normalise the explanations and calculate the Euclidean distances between the instances in both clusters. Smaller mean pairwise distance = better explainability framework metric.

5. **Computational Efficiency**. Average time taken, in seconds, by the interpretability framework to output a set of explanations. (Similar Cloud environments are applied to all experiments).

A metric such as *'Computational Efficiency'* could be considered unrelated to a measure of explainability, but this research proposal contends that it is an important value to consider in terms of feasibility/practicality of XAI methods. Computational time can be a bottleneck in generating explanations and may have an impact on the commercial viability of an explainability process in a credit card fraud detection application.

This will be a deductive approach to test the assumption that one particular interpretability framework can be shown, through statistical significance testing on the numerical outputs of each experiment, to generate the best local explanations for a credit card fraud classification result. Although the experiments of (Evans et al., 2019) focused on global explanations, their experiments used a Friedman test to collate the p-values into a correlation matrix, and although the metrics used are different from those proposed in this paper, this is a general approach that will be partially imitated.

A Friedman test will be run to determine whether there is evidence that there is a statistical difference in performance between SHAP, LIME, ANCHORS, and DiCE in terms of explaining local credit card fraud classification results. The research assumption will be that a calculated P-value of less than 0.05 implies that a given technique can be ranked higher than the others or not.

A subsequent Wilcoxon signed rank test would be run on each pair of interpretability techniques to measure the degrees of separation based on the custom metrics.

Chapter 4

Results, evaluation and discussion

4.1 The Build and Evaluation of CC Fraud NN Detection Model

4.1.1 Building the Credit Card Fraud Model

The credit card transaction data set, labelled *non-fraud (0)* and *fraud (1)*, used in this thesis originated in a demo application of a working product. It contains a number of columns used for ETL and other custom integration purposes, which are redundant for the purposes of fraud classification. Eliminating these data elements reduced the dimensionality of the dataset from 380 columns to 263.
In order to accelerate the model building process, the AutoML engine in Azure ML Studio (https://studio.azureml.net/) was used on the refined dataset to rank the features in order of importance for the classification result.
It is difficult to generate meaningful XAI explanations on highly dimensioned data, so a balance was struck between reducing the dataset size, maintaining accuracy, and producing worthwhile XAI output.

The primary considerations in the initial model building process were;

1. Limiting the feature set to the top 40 columns that would still produce a Neural Network predictive fraud model with 85%+ accuracy.

2. The experiments described in Section 4.2.4 for DiCE explanations initially performed very poorly because the subset of features did not have a sufficiently wide range of continuous data. Increasing the feature set by adding another 10 ranking columns, which contained transaction amount data with a greater depth of variation, greatly improved the quality of DiCE explanations.

3. Anchor explanations, as generated in the experiments described in Section 4.2.3, also benefited slightly from the addition of more columns in the model building process.

4. Through an iterative model building process, the model accuracy and the F1 score could be maintained with an increased feature set of 55 columns (see the image below).

RANK	FEATURE	RELEVANCE	TYPE	FIELD_ID	FUNCTION
2	NonEMVTransactionsCount.cnt.day.present	0.1147	continuous	146	predictor
3	MerchantCategory	0.1082	continuous	13	predictor
4	POS_Count.cnt.day.present	0.1076	continuous	282	predictor
5	PinIndicator	0.1053	categorical	71	predictor
6	DomesticAuthCount.cnt.hour1	0.1041	continuous	220	predictor
7	DomesticAuthCount.cnt.hour3	0.1026	continuous	221	predictor
8	DomesticAuthCount.cnt.hour4	0.1022	continuous	222	predictor
9	DomesticAuthCount.cnt.hour10	0.0963	continuous	223	predictor
10	DomesticAuthCount.cnt.hour15	0.0959	continuous	224	predictor
11	DomesticAuthCounter.cnt.day.present	0.0951	continuous	167	predictor
12	DomesticAuthCount.cnt.hour25	0.0931	continuous	225	predictor
13	OnlinePOSCountForever.cnt.present	0.0866	continuous	247	predictor
14	POSTerminalAttendedAuthCount.cnt.day.present	0.0859	continuous	188	predictor
15	CustomerNotPresentAuthCount.cnt.day.present	0.0845	continuous	241	predictor
16	DvcVerificationCap	0.0824	continuous	45	predictor
17	ECommerceAuthCount.cnt.day.present	0.0783	continuous	207	predictor
18	PosTerminalAttended	0.0777	categorical	48	predictor
19	TxnChannelCode	0.0773	categorical	4	predictor
20	CustomerPresentIndicator	0.0768	categorical	47	predictor
21	OnlineNewMerchCtryCntDaily.cnt.day.present	0.0736	continuous	202	predictor
22	OnlineNewMerchCtryCntHourly.cnt.hour24	0.0735	continuous	86	predictor
23	OnlineNewMerchCtryCntHourly.cnt.hour15	0.0732	continuous	85	predictor
24	OnlineNewMerchCtryCntHourly.cnt.hour10	0.0729	continuous	84	predictor
25	DvcPosEntryMode	0.0717	categorical	44	predictor
26	OnlineNewMerchCtryCntHourly.cnt.hour3	0.0713	continuous	82	predictor
27	OnlineNewMerchCtryCntHourly.cnt.hour4	0.0709	continuous	83	predictor
28	OnlineNewMerchCtryCntDaily.cnt.day.total	0.0707	continuous	201	predictor
29	NotECommerceAuthCount.cnt.day.present	0.0691	continuous	227	predictor
30	NewMerchantCountryCount.cnt.hour15	0.0687	continuous	345	predictor
31	OnlineNewMerchCtryCntHourly.cnt.hour1	0.0687	continuous	81	predictor
32	NewMerchantCountryCount.cnt.hour10	0.068	continuous	344	predictor
33	NewMerchantCountryCount.cnt.hour24	0.0679	continuous	346	predictor
34	ECommerceFlag	0.0662	categorical	66	predictor
35	NewMerchantCountryCount.cnt.hour4	0.0651	continuous	343	predictor
36	NewMerchantCountryCount.cnt.hour3	0.0648	continuous	342	predictor
37	AuthResponse	0.0642	categorical	27	predictor
38	NewMerchantCountryCount.cnt.hour1	0.0621	continuous	341	predictor
39	AmountBase	0.0575	continuous	34	predictor
40	CardType	0.056	categorical	38	predictor
41	POSSum.acc.month.total	0.056	continuous	102	predictor
42	NotECommerceAuthAmount.acc.day.total	0.056	continuous	125	predictor
43	NonEMVTransactionsAcc.acc.day.total	0.056	continuous	152	predictor
44	POSTerminalAttendedAuthAmount.acc.day.total	0.056	continuous	87	predictor
45	CustomerPresentAuthAmount.acc.day.total	0.056	continuous	354	predictor
46	EMVTransactionsAcc.acc.day.total	0.056	continuous	334	predictor
47	CustomerNotPresentAuthAmount.acc.day.total	0.056	continuous	173	predictor
48	HourlyAuthAmt.acc.hour25	0.056	continuous	137	predictor
49	NonEMVTransactionsAcc.acc.day.present	0.056	continuous	153	predictor
50	NotECommerceAuthAmount.acc.day.present	0.056	continuous	126	predictor
51	CustomerNotPresentAuthAmount.acc.day.present	0.056	continuous	174	predictor
52	POSTerminalAttendedAuthAmount.acc.day.present	0.056	continuous	88	predictor
53	CustomerPresentAuthAmount.acc.day.present	0.056	continuous	355	predictor
54	HighRiskPOSSum.acc.hour.total	0.056	continuous	372	predictor
55	EMVTransactionsAcc.acc.day.present	0.056	continuous	355	predictor
56	Fraud	0	categorical	72	label

Figure 4.1: Final Feature List for Predictive Credit Card Fraud Model

5. As the objective was to build a predictive model with a Neural Network algorithm a Tensor Flow *Keras* library was used in the Python Notebook

model creation process. The *keras-tuner* library was used to set a series of hyperparameter tuning options to optimise model creation.

The table above does not reflect the subsequent feature engineering that applied *one hot encoding* to the categorical features before model building.

Appendix B of this thesis also provides a number of visual descriptions of key features remaining within the credit card fraud dataset.

4.1.2 Evaluating the Predictive Fraud Model

The predictive credit card fraud model used in these experiments was required to demonstrate strong performance metrics to ensure that meaningful XAI output data was generated.

Model Evaluation Metrics

After a series of iterations, including refinements required to improve the quality of the XAI experiment results in Section 4.2, a credit card predictive model was created with the evaluation metrics in the table figure 4.1 below.

Table 4.1: Model Performance Metrics

Accuracy	0.863775
ROC AUC Score	0.939888
Precision (Class 0)	0.926186
Recall (Class 0)	0.793675
F1-Score (Class 0)	0.854826
Precision (Class 1)	0.816107
Recall (Class 1)	0.935385
F1-Score (Class 1)	0.871685

The **Accuracy** and **F1-Score (Class 1 - 'Fraud')** scores are the primary metrics for which the acceptable threshold of 85% was established and achieved.

Loss Function Graph

The loss function graph is a fundamental component in evaluating the performance of a machine learning model, especially in the context of neural networks used for binary classification tasks such as credit card fraud detection. The loss function measures the errors made by the model during training. The graph of this function plots these error values over the course of the training epochs for the fraud classification model. In the context of our model, which has demonstrated high Accuracy and F1-Score, the loss function graph serves as a crucial additional diagnostic tool, providing insights into how well the model learned from the training data over time.

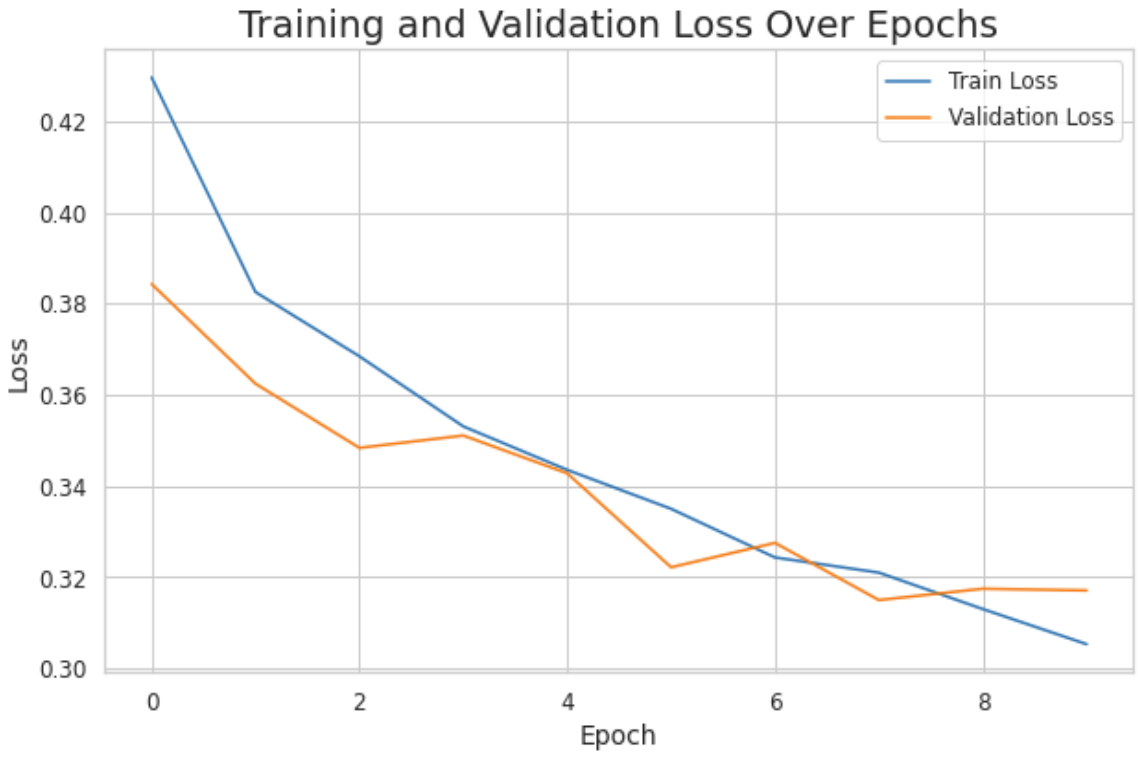


Figure 4.2: Loss Function Graph to Evaluate Performance of CC Predictive Model

As the epochs progress, both the training and the validation loss exhibit a downward trend, with the training loss decreasing from over 0.42 to just over 0.30, and the validation loss also reducing to just under 0.32 after eight epochs. This decrease indicates that the model is learning from the training data and improving its

predictive accuracy. In particular, the validation loss intersects with the training loss at the 4th epoch and twice thereafter. These convergence points are significant: it suggests that the model, at this stage, is performing equally well on both the training data and unseen data from the validation set. This scenario indicates a good generalisation in our fraud classification model and a suitable starting point for the XAI metrics experiments.

Confusion Matrix

The Confusion Matrix below in figure 4.3 provides a visualisation of the prediction accuracy.

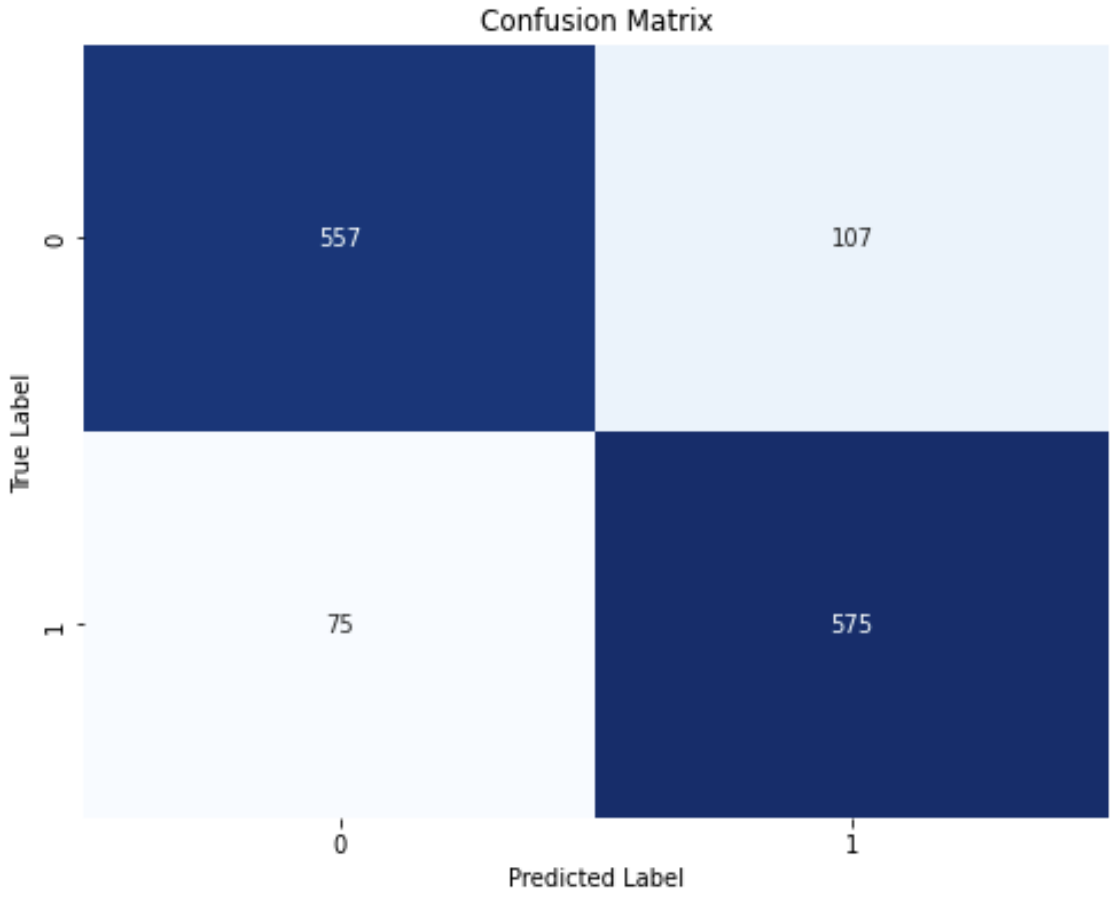


Figure 4.3: Confusion Matrix for CC Model Evaluation

4.2 Results of the XAI Metrics Experiments

As described in Section 3.2.3 of this paper, the experiments to generate the metrics for each XAI explainer followed this sequence of steps:

1. Pre-build the optimised Neural Network model on the research credit card dataset and store within the Kubeflow environment.
2. Verify each implementation of the XAI Python methods on a small subset of records from the test data (˜5 rows of data).
3. Divide the entire test data set into 20 consecutive *'chunks'* of equal size.
4. For each individual XAI explainer generate explanations for the rows in each test data *'chunk'*
5. Calculate the five metrics scores/values for each data *'chunk'*
6. Store the metrics to file once generated for each *'chunk'*. Repeat until all *'chunks'* have been processed, and the results are stored on the Kubeflow file system.
7. For each metric, collate the scores (based on the four XAI methods) of the 20 iterations of the experiment and use these values as input to a statistical analysis exercise.
8. Additionally, for each metric, calculate the mean score (based on the XAI method) of the 20 iterations of the experiment. This will provide further insight into the relevant strengths and weaknesses of the XAI techniques.
9. Perform a statistical analysis to support either the NULL or Alternate hypothesis of this paper.

4.2.1 SHAP: XAI Experiment Results

Verification of SHAP Explainer

Each of the four experiments on the machine learning interpretability techniques in this thesis began with a verification that the explainers actually produce meaningful output. Both the NN classification model and the data from the test set must generate an explanation that provides useful information about the classification result.

This thesis follows the objective of defining a purely statistical approach to evaluating XAI methods at scale, but a visual (manual) inspection is carried out first to confirm the validity of each XAI method. More specifically, the first step of each experiment is to ensure that the actual implementation of the Python XAI libraries is working as expected.

The preliminary check of the SHAP values involves a summary plot generated from the first 25 rows in the test data set. The SHAP explainer is created based on the NN Credit Card predictive model built in a previous step, along with a selection of training data. The *shap.summary_plot()* function is used to display the top twenty-five features that influence the classification of *non-fraud*, as shown below in figure 4.4.

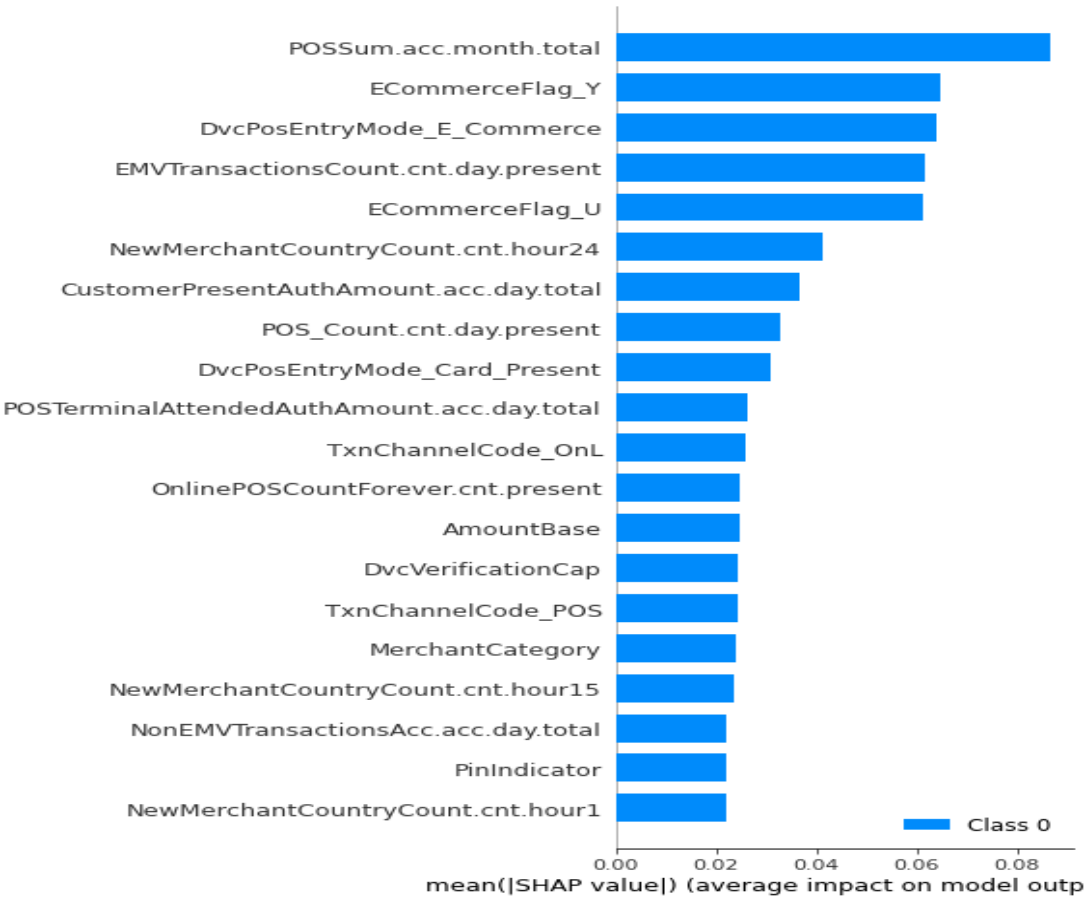


Figure 4.4: SHAP Summary Plot - Based on first 25 rows in test dataset

Looking at a single instance, selected at random from the test dataset, a set of SHAP values can be generated for each feature.

Top 20 Features and Their SHAP Values:

Feature	SHAP Value
MerchantCategory	0.056838
DvcPosEntryMode_E_Commerce	0.055646
EMVTransactionsCount.cnt.day.present	0.044492
ECommerceFlag_Y	0.041965
POSSum.acc.month.total	0.040538
ECommerceFlag_U	0.035380
PinIndicator	0.032996
DvcPosEntryMode_Card_Present	0.030514
TxnChannelCode_OnL	0.030263
NewMerchantCountryCount.cnt.hour24	0.027834
TxnChannelCode_POS	0.024505
NewMerchantCountryCount.cnt.hour1	0.023687
NewMerchantCountryCount.cnt.hour15	0.020875
POSTerminalAttendedAuthAmount.acc.day.present	0.018786
CustomerPresentAuthAmount.acc.day.total	0.018662
DomesticAuthCount.cnt.hour25	0.018191
OnlineNewMerchCtryCntHourly.cnt.hour15	0.017986
CustomerPresentIndicator_Y	0.017196
CustomerNotPresentAuthCount.cnt.day.present	0.016722
OnlinePOSCountForever.cnt.present	0.015733

Figure 4.5: Random Instance: SHAP Score of Top 20 Features Determining Classification

The figure above (4.5) shows the score generated for the key features contributing to the classification of this one instance.

SHAP: Pre-Experiment Adjustments/Considerations

The output of the aggregate SHAP values from the *shap.summary_plot()* function and the SHAP values assigned to a single random fraud instance demonstrate that the Python SHAP library is able to process the credit card data set for this research. No additional pre-processing is required and it is possible to proceed to the scoring of the SHAP values for the test data using the custom metrics defined for this thesis.

SHAP Output Results from Metric Scoring

The SHAP explainer experiments produced the following table of results. Each sample row in the table in figure 4.6 represents the metrics calculated on a sequential block from the test dataset.

Sample Number	XAI_Identity	XAI_Stability	XAI_Seperability	XAI_Similairity	Comp_Efficiency
1	40.0000	87.6923	96.9231	0.2746	384.51
2	46.1538	76.9231	96.9231	0.2320	386.19
3	49.2308	67.6923	90.7692	0.2071	384.94
4	36.9231	75.3846	93.8462	0.2637	386.69
5	47.6923	83.0769	100.0000	0.2800	384.34
6	44.6154	27.6923	93.8462	0.2144	391.56
7	36.9231	75.3846	100.0000	0.3984	386.00
8	38.4615	13.8462	100.0000	0.2623	390.81
9	46.1538	70.7692	96.9231	0.3450	393.99
10	43.0769	33.8462	96.9231	0.3944	390.54
11	40.0000	84.6154	100.0000	0.2189	397.76
12	41.5385	32.3077	100.0000	0.3294	398.10
13	38.4615	70.7692	100.0000	0.2444	399.48
14	27.6923	80.0000	100.0000	0.2967	396.85
15	32.3077	75.3846	95.3846	0.3791	394.65
16	35.3846	76.9231	95.3846	0.3148	390.70
17	43.0769	29.2308	100.0000	0.2203	392.06
18	53.8462	29.2308	96.9231	0.2233	386.87
19	43.0769	23.0769	93.8462	0.2148	381.84
20	41.5385	46.1538	100.0000	0.3053	387.78

Figure 4.6: SHAP XAI Experiment: Metrics Scores

Explaining the XAI Metric Scorecard (for All XAI Methods)

- The Sample Number identifies the individual data *chunk* extracted from the test dataset.
- *XAI Identity* is the separate score obtained from each data *chunk*. This is a score that can range from zero to 100.
- *XAI Stability* is also a score in the range of zero to 100 for each data *chunk*.
- *XAI Seperability* is another score of 0 - 100. In practice, the definition of 'duplicate' required that this metric allow for a small threshold/tolerance when determining whether instance explanations were alike.

- *XAI Similarity* is a Euclidean measure of the average distance between points scored for this metric for each data *chunk*.

- *Computational Efficiency* is the time taken in seconds for the XAI method to actually generate explanations for each data *chunk*.

The mean of each set of column values is used as input to the statistical analysis described in Section 4.1.2.

SHAP: Initial Experiment Observations

The Python SHAP library generates SHAP (SHapley Additive exPlanations) values for each feature in a test dataset, quantifying the contribution of each feature to the prediction for each individual instance. These values offer an interpretable measure of the importance of the features.
The output is in numerical format and therefore did not require any arithmetical conversion before being processed by the XAI Metric functions in this dissertation. We could consider these SHAP metrics scores as an effective *baseline* for the sequence of experiments in this research, given the relative ubiquity of SHAP values in ML Financial Services crime detection products.

4.2.2 LIME XAI Experiments: Results

Verification of LIME Explainer

The Python LIME (Local Interpretable Model-agnostic Explanations) library generates explanations for individual predictions of any classifier or regressor by approximating the model locally around each data point. For each feature in a test dataset, LIME provides interpretable insights by presenting a simplified model that captures the local behaviour of the complex model, highlighting which features were most influential in the specific instance's prediction.
Following the established steps for these XAI metrics experiments, a random instance was selected from the test data to verify the output of the *lime_tabular()*

Python function. The figure below (4.7) represents the Notebook display of the LIME explanation of a single instance.

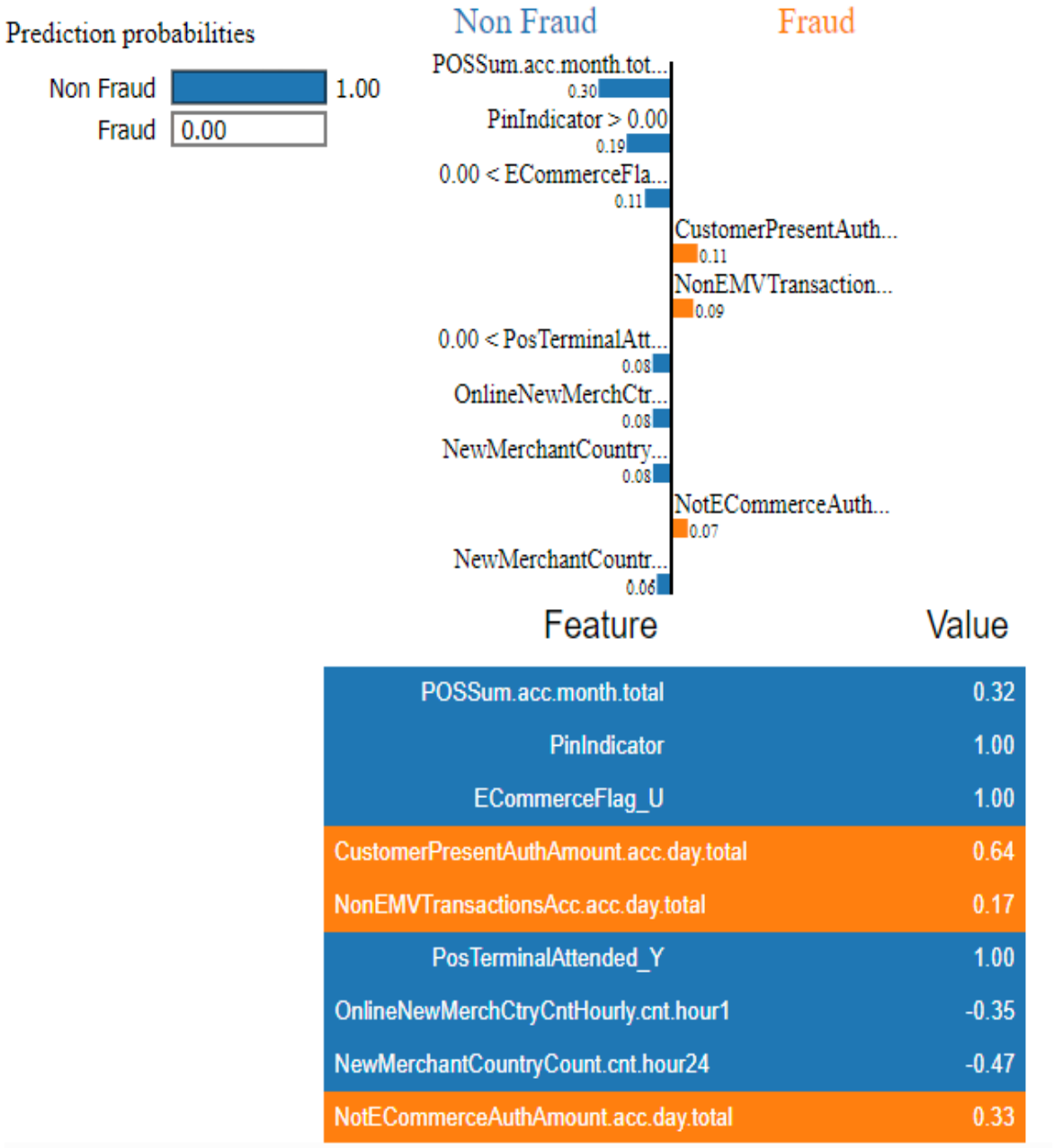


Figure 4.7: Random Instance: LIME Explainers

The features on the left-hand side, for example the *PinIndicator*, represent those attributes in the given transaction that most contribute to the classification of *Non-Fraud*.

Another view of LIME explanations generated on the credit card fraud dataset is represented in Figure 4.8.

```
1/1 [==============================] - 0s 17ms/step
157/157 [==============================] - 0s 1ms/step


----------------------------------------
Real Value: Fraud    1
Name: 1, dtype: int64
Predicted Probability (Fraud): 0.91

Features in Order of Importance:
Feature: PinIndicator | Value: <= 0.00 | Weight: 0.20
Feature: POSSum.acc.month.total | Value: <= -0.63 | Weight: 0.12
Feature: ECommerceFlag_U | Value: <= 0.00 | Weight: 0.12
Feature: EMVTransactionsCount.cnt.day.present | Value: <= -0.51 | Weight: 0.09
Feature: PosTerminalAttended_Y | Value: <= 0.00 | Weight: 0.08
Feature: POS_Count.cnt.day.present | Value: <= -0.51 | Weight: -0.07
Feature: AuthResponse | Value: <= -0.22 | Weight: -0.07
Feature: 0.00 | Value: < DvcPosEntryMode_E_Commerce <= 1.00 | Weight: -0.06
Feature: CustomerPresentAuthAmount.acc.day.total | Value: <= -0.62 | Weight: -0.06
Feature: OnlineNewMerchCtryCntHourly.cnt.hour1 | Value: > -0.35 | Weight: 0.06
Feature: DvcPosEntryMode_Card_Present | Value: <= 0.00 | Weight: -0.06
Feature: -0.55 | Value: < NotECommerceAuthAmount.acc.day.total <= -0.33 | Weight: -0.05
Feature: 0.00 | Value: < ECommerceFlag_Y <= 1.00 | Weight: 0.04
Feature: -0.18 | Value: < CustomerNotPresentAuthAmount.acc.day.present <= -0.11 | Weight: -0.04
Feature: -0.52 | Value: < POSTerminalAttendedAuthAmount.acc.day.total <= -0.30 | Weight: 0.04
```

Figure 4.8: Fraud Instance: LIME Explainers

For simplicity, the explanations have been limited to the top 15 features, in order of importance. This is another instance randomly selected from the test data, but a record classified as *Fraud* by the model, which matches the 'real' classification for the record. The numerical values in Figure 4.8 represents the value of the features after scaling. In practice, non-binary values would be reverse scaled to increase the meaningfulness of the explanation.

In Figure 4.8, the predicted probability of a fraudulent transaction is high, which means that the model recognises patterns in this instance similar to other fraudulent instances from the training data. Features with positive weights contributed to increasing the probability of a fraud prediction. The larger the weight, the more influential the feature. In contrast, the negative weighting characteristics worked against the fraud prediction. The higher the negative weight, the more it tried to

reduce the probability.

For each feature displayed, the *Value* tells you what the specific value of that feature was for the instance, and the *Weight* tells you how much that feature influenced the prediction. By examining the top features and their weights, you can get a good understanding of why the model made its prediction.

LIME: Pre-Experiment Adjustments/Considerations

As with the SHAP experiments, it has been established that the Python LIME library, *lime_tabular()*, will generate meaningful output on the credit card fraud dataset. LIME explanations are returned in order of weight value, with the highest being generated and displayed first. A practical observation in the XAI pre-experiment phase for LIME was that the weights returned for features became increasingly insignificant after the first 25% of values were returned. Therefore, by setting *number_features = 16* in the Python code, the LIME explanations were limited to the 16 features most influential for the prediction. This simplified the prediction/explanation process without any real compromising of the explanation.

The LIME explanations are returned as a <*feature*> <*weight*> combination. For every feature with a LIME-generated weight, this was assigned as input to the XAI metrics functions. Those features that did not generate a weight, based on the thresholds described above, were assigned a value of zero.

LIME Output Results from Metric Scoring

Each sample row in the table in Figure 4.9 below represents the LIME metrics calculated on a sequential block from the test dataset.

Sample Number	XAI_Identity	XAI_Stability	XAI_Seperability	XAI_Similairity	Comp_Efficiency
1	10.8214	26.2323	100.0000	0.6541	754.96
2	3.0769	78.4615	90.7692	0.5767	766.21
3	3.0769	60.0000	100.0000	0.5457	726.49
4	4.6154	33.8462	93.8462	0.6235	749.68
5	12.3077	40.0000	100.0000	0.6593	740.32
6	1.5385	24.6154	96.9231	0.5396	724.98
7	3.0769	69.2308	90.7692	0.5989	733.20
8	4.6154	49.2308	100.0000	0.6556	754.26
9	10.7692	26.1538	100.0000	0.6376	744.94
10	4.6154	35.3846	100.0000	0.6671	730.45
11	3.0769	66.1538	100.0000	0.5025	732.48
12	3.0769	63.0769	93.8462	0.6177	731.88
13	3.0769	81.5385	93.8462	0.5432	742.28
14	1.5385	67.6923	100.0000	0.5480	728.85
15	3.0769	26.1538	93.8462	0.6075	728.72
16	0.0000	73.8462	100.0000	0.6395	727.66
17	6.1538	40.0000	96.9231	0.5484	730.83
18	4.6154	32.3077	90.7692	0.5875	739.91
19	6.1538	36.9231	96.9231	0.4402	745.11
20	3.0769	64.6154	96.9231	0.6025	729.68

Figure 4.9: LIME XAI Experiment: Metrics Scores

LIME: Initial Experiment Observations

The *Identity* metric measures how many identical instances (or close to identical) have identical explanations. LIME scores poorly in experiments for this measure. The simplified mode used by the LIME algorithm appears to produce more variance in the explanations than is seen with SHAP values.

4.2.3 ANCHORS XAI Experiments: Results

Verification of Anchor Explainer

In the context of Explainable AI (XAI), the Python ANCHOR library generates feature-specific explanations for individual instances in a test dataset by identifying minimal sets of conditions, or 'anchors', that are sufficient to ensure the same prediction for similar instances. Each anchor explanation highlights the features and their respective values that are most influential in the model's decision-making

process for that particular instance, providing a local, instance-based understanding of the model's behaviour.

Again, random instances were selected from the test data to verify the output of the *anchor_tabular()* Python function.

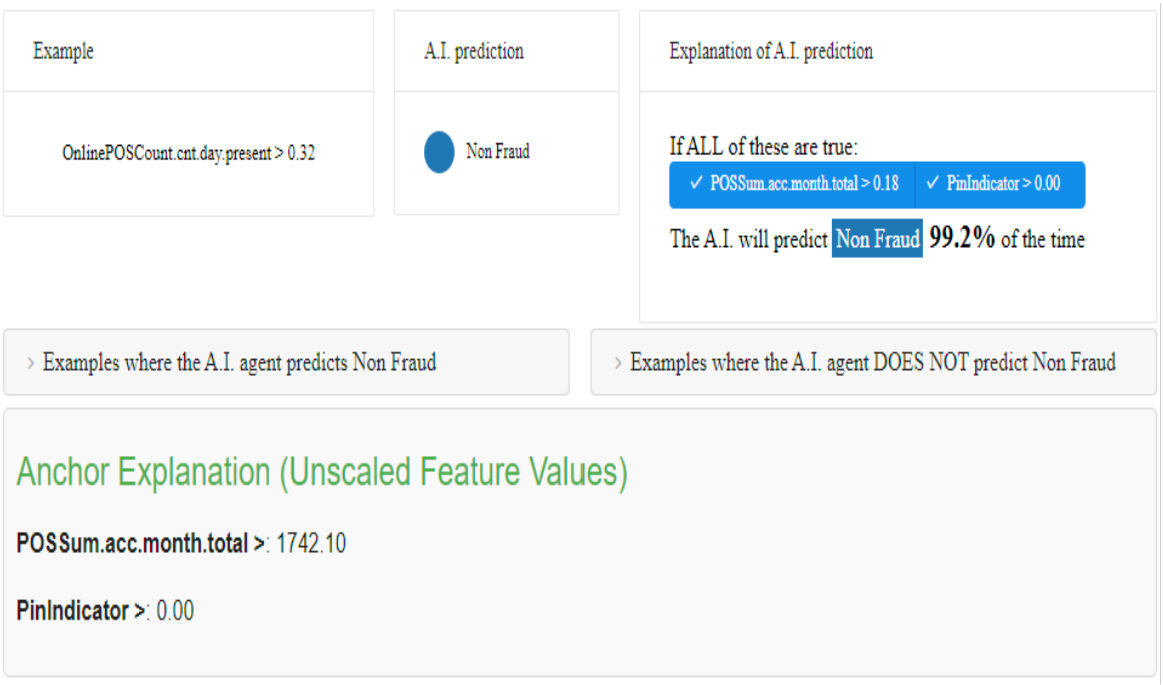


Figure 4.10: Non-Fraud Instance: ANCHOR Explainers

In the above figure (4.10) the 'Explanation of A.I. prediction' is a set of conditions that constitute the *anchor*. These are the specific feature values or ranges for the instance being explained that, when fixed, provide information on the characteristics and values that are most crucial to this prediction of *Non-Fraud*.

The model and the ANCHOR explainer work with the same set of scaled input data. An inverse transform has been generated in each diagram to add additional meaning by converting the values back to their original scale. In this previous example, instances will always be predicted as *Non-Fraud* when the transaction is part of a sequence of transactions at the same POS machine that total more than $1742 and a PIN was entered.

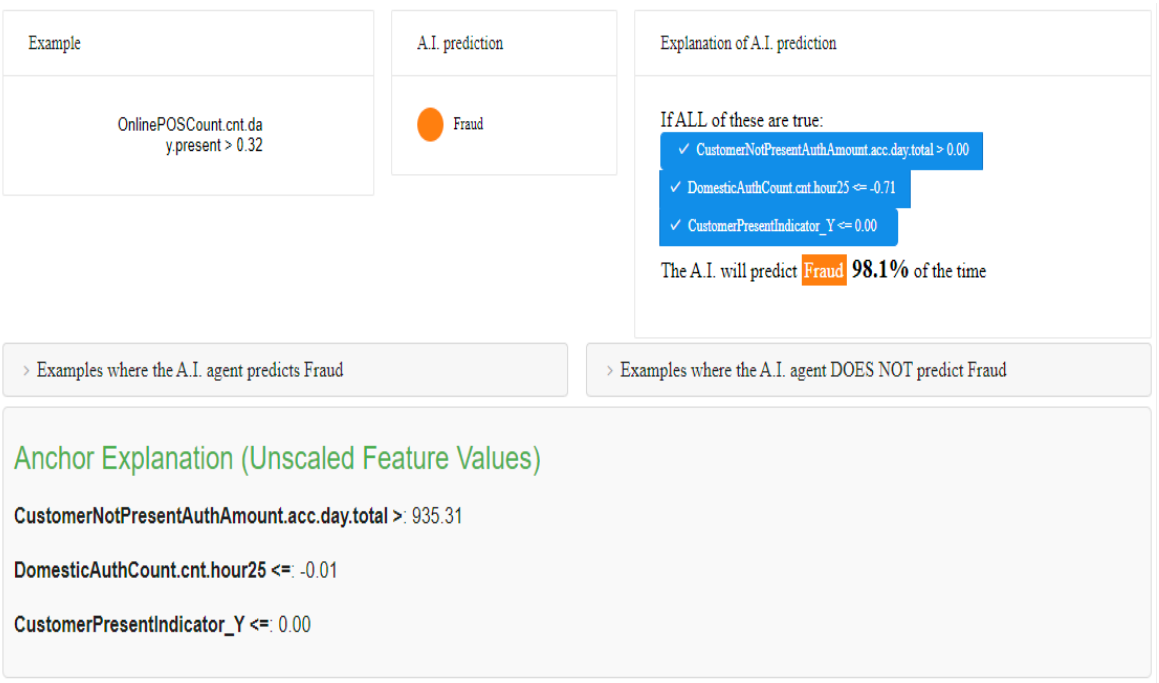


Figure 4.11: Fraud Instance: ANCHOR Explainers

In the above figure (4.11), the anchors show the conditions that will always result in a *Fraud* prediction.

Anchor: Pre-Experiment Adjustments/Considerations

The examples in the previous section have verified that the *anchor_tabular()* library function produces meaningful explanations for the credit card dataset used in this research.

The Anchor explanations for each instance are parsed in sequence to create a row in a new dataframe. The feature columns identified as 'anchors' are populated with the numerical value of that weight. The other ('non-anchor') feature columns are populated with a default value of '-1'.

Anchor Output Results from Metric Scoring

The ANCHORS explainer experiments produced the following table of results. Each sample row in the table represented in figure 4.12 represents the metrics calculated on a sequential block from the test dataset.

Sample Number	XAI_Identity	XAI_Stability	XAI_Seperability	XAI_Similairity	Comp_Efficiency
1	15.6250	82.8125	15.6250	1.5377	2901.37
2	12.5000	60.9375	20.3125	1.1226	2783.96
3	6.2500	60.9375	25.0000	1.7104	2799.25
4	9.3750	32.8125	20.3125	1.4956	2803.50
5	6.2500	20.3125	18.7500	1.7493	2726.46
6	12.5000	43.7500	17.1875	1.0614	3072.32
7	4.6875	29.6875	25.0000	2.1536	2903.02
8	7.8125	60.9375	32.8125	1.2250	2713.01
9	1.5625	26.5625	23.4375	1.7183	2796.82
10	9.3750	53.1250	17.1875	2.0590	3005.47
11	9.3750	65.6250	15.6250	1.0507	2647.00
12	12.5000	45.3125	20.3125	1.7524	3148.17
13	18.7500	60.9375	25.0000	1.6294	2722.98
14	9.3750	73.4375	20.3125	1.8468	2672.22
15	3.1250	28.1250	18.7500	2.0189	2627.63
16	14.0625	43.7500	17.1875	1.5749	2791.43
17	7.8125	40.6250	25.0000	1.6403	2900.18
18	12.5000	43.7500	32.8125	2.3355	2914.33
19	10.9375	60.9375	23.4375	1.0377	2735.07
20	9.3750	64.0625	17.1875	1.8237	2994.62

Figure 4.12: ANCHOR XAI Experiment: Metrics Scores

ANCHORS: Initial Experiment Observations

The *Computational Efficiency* score for Anchor explanations is noticeably higher (worse) for this XAI method, when compared to the other explainer approaches. However, during the execution of the actual Anchor XAI metrics experiment, it was observed that within each data block, it was generally only approximately 5% of the records that consumed the majority of the processing time.

The Python Notebook generated a textual representation of the Anchor explanations as each feature was processed. For most records, the Anchor length was limited to two or three features, with associated numerical relationships, typically in the format `<feature> <relationship> <value>` , for example;

$$NonEMVTransactionsAcc.acc.day.total > 75.00 \tag{4.1}$$

The Python *anchor_tabular* library provides a *threshold* parameter to limit the complexity of the Anchor explanation generated for each feature. However, even

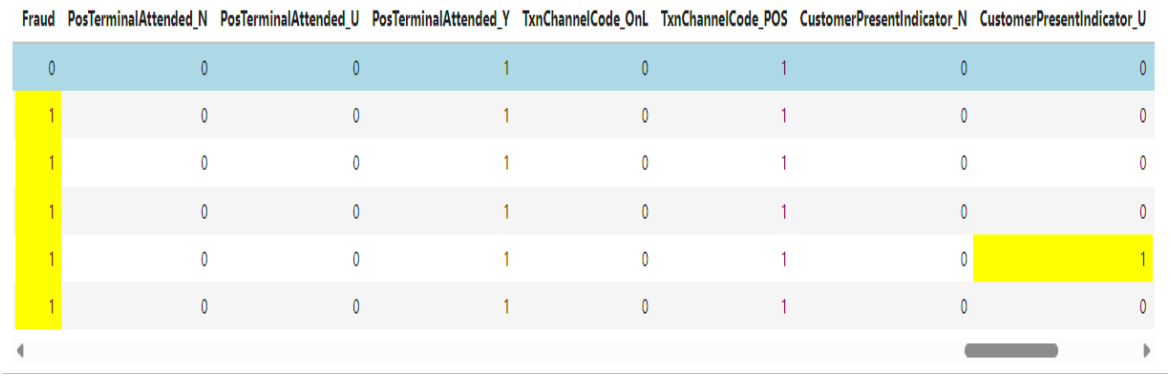
increasing the setting of this value to *0.99* did not prevent a small number of instances from becoming outliers in terms of computational processing expense. The *Identity* metric also scores poorly. Similar instances produce a greater diversity of Anchors than is seen with other interpretability techniques.

4.2.4 DiCE XAI Experiments: Results

Verification of DiCE Explainer

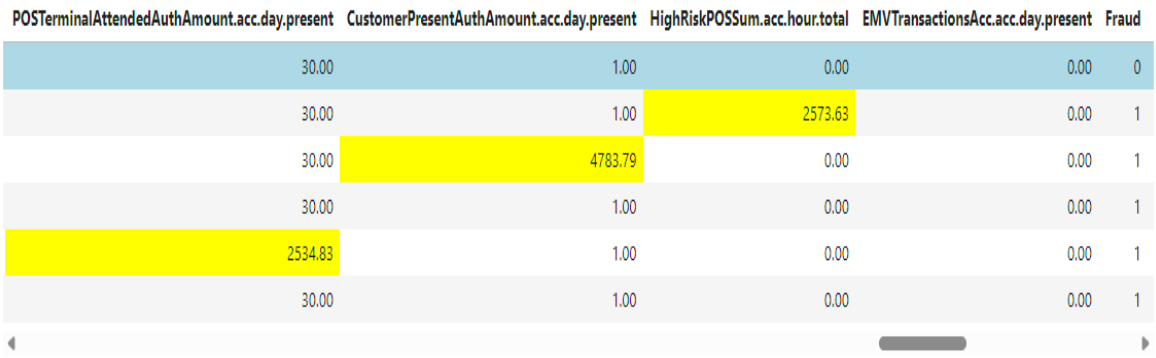
The Python DiCE library generates counterfactual explanations for individual instances in a test dataset, focussing on identifying minimal changes to the feature values that would alter the model's prediction. For each feature of a given instance, DiCE provides alternative scenarios, presenting how adjusting the feature values could lead to a different decision from the model, thereby offering insights into the model's sensitivity and decision boundaries.
In line with previous experiments in this thesis, a random instance of the test data was selected to assert that the *dice_ml* Python libraries generated meaningful explanations on the credit card dataset.
To enhance the visual inspection of the counterfactuals for a given instance, a custom Python display function was written to highlight the generated DiCE output. The figures below show a subsection of the instance for which five counterfactual lines were generated. The first line (shaded) is the actual instance attributes. Under this line are the counterfactuals generated. The highlighted cells represent changes in the value of the attribute that would switch the classification prediction for the instance.

In Figure 4.13 the inability to identify if a customer was present at a credit card transaction would be one of the counterfactual conditions that alters the original prediction from *Non-Fraud* to *Fraud*.



Fraud	PosTerminalAttended_N	PosTerminalAttended_U	PosTerminalAttended_Y	TxnChannelCode_OnL	TxnChannelCode_POS	CustomerPresentIndicator_N	CustomerPresentIndicator_U
0	0	0	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	0	1	0	1	0	1
1	0	0	1	0	1	0	0

Figure 4.13: Fraud Instance: DiCE Explainer Label + Customer Present Indicator



POSTerminalAttendedAuthAmount.acc.day.present	CustomerPresentAuthAmount.acc.day.present	HighRiskPOSSum.acc.hour.total	EMVTransactionsAcc.acc.day.present	Fraud
30.00	1.00	0.00	0.00	0
30.00	1.00	2573.63	0.00	1
30.00	4783.79	0.00	0.00	1
30.00	1.00	0.00	0.00	1
2534.83	1.00	0.00	0.00	1
30.00	1.00	0.00	0.00	1

Figure 4.14: Fraud Instance: DiCE Explainer Label + Counterfactual Transaction Amounts

Similarly, an increase in the *POSTerminalAttendedAuthAmount.acc.day.present* **or** *CustomerPresentAuthAmount.acc.day.present* **or** *HighRiskPOSSum.acc.hour.total* values beyond the thresholds shown in Figure 4.14 will also change the classification prediction to *Fraud*.

DiCE: Pre-Experiment Adjustments/Considerations

The DiCE counterfactuals did **not** generate on the first pass through the experiment. The initial feature engineering process reduced the set of attributes to the top 40 most important columns necessary to build a performant predictive fraud model. However, this subset did not have enough variance in the continuous data values to allow the *dice_ml* library generate any counterfactual explanations. It was necessary

to re-build the predictive NN model with new features containing extra information on the transaction amounts recorded in the hours/days on either side of the actual instance transaction. Once these additional attributes were included in the model build process, it was possible to generate the counterfactuals.
A selection of these counterfactuals is presented in the previous subsection and was sufficient validation to allow progression to the XAI metrics experiments.
For simplicity, only the first counterfactual row generated for an instance was considered and sequentially converted into input for the XAI metrics functions.

DiCE Output Results from Metric Scoring

The DiCE explainer experiments produced the following table of results. Each sample row in Table 4.15 represents the metrics calculated on a sequential block from the test dataset.

Sample Number	XAI_Identity	XAI_Stability	XAI_Seperability	XAI_Similairity	Comp_Efficiency
1	6.1538	58.4615	35.3846	13.1316	76.47
2	18.4615	46.1538	36.9231	16.2164	77.33
3	9.2308	56.9231	41.5385	13.8937	122.52
4	24.6154	55.3846	40.0000	14.9150	75.33
5	7.6923	55.3846	49.2308	13.9490	121.45
6	20.0000	43.0769	49.2308	19.9329	79.69
7	15.3846	60.0000	52.3077	19.7095	131.22
8	13.8462	55.3846	44.6154	12.3597	75.06
9	18.4615	47.6923	49.2308	14.6446	76.56
10	6.1538	50.7692	46.1538	18.9109	84.86
11	12.3077	64.6154	53.8462	15.0291	83.61
12	10.7692	66.1538	49.2308	15.7752	79.67
13	16.9231	58.4615	40.0000	19.0540	89.36
14	7.6923	73.8462	53.8462	14.8314	75.93
15	9.2308	64.6154	50.7692	16.7801	82.30
16	9.2308	56.9231	49.2308	15.1260	76.56
17	7.6923	61.5385	40.0000	16.5128	76.23
18	13.8462	58.4615	58.4615	16.9805	76.22
19	24.6154	44.6154	44.6154	13.7436	80.85
20	4.6154	60.0000	55.3846	18.4578	78.25

Figure 4.15: DiCE XAI Experiment: Metrics Scores

DiCE: Initial Experiment Observations

Counterfactuals do not score well in the Identity metric because the size of a counterfactual will often need to be quite large to 'flip' a fraud classification for a given instance.
Thus, the Euclidean distance values between different XAI instances can be quite large. Instances that are similar to each other can still have a greater variation in terms of the magnitude of generated Counterfactual values.

4.2.5 Aggregate XAI Experiment Results

Taking the output generated by the XAI metrics experiments produces the following matrix of mean values, as shown in Table 4.2.

Table 4.2: Final Table of XAI Metrics Results

	SHAP	**LIME**	**ANCHORS**	**DiCE**
Identity	41.308	4.618	9.688	12.846
Stability	58.000	49.773	49.922	56.923
Separability	97.385	96.769	21.563	47.000
Similarity	0.281	0.590	1.627	15.998
Computational Efficiency	390.282	738.145	2832.941	85.974

4.3 Evaluation of XAI Metrics Results

4.3.1 Friedman Test Analysis

Tabular View of Friedman Results

The Friedman test was an appropriate choice for evaluating the NULL hypothesis in this study, as it is a non-parametric test ideal for comparing multiple groups (XAI methods in this case) across different measures without assuming a normal distribution of the data. It compares the ranks of the scores across different groups,

and this test's robustness against non-normal distributions makes it particularly suitable for the heterogeneous nature of data commonly encountered in XAI method comparisons in Data Science research.

Given that the custom XAI metrics from the research experiments comprises different measures applied to each XAI method, the Friedman test effectively evaluates that the NULL Hypothesis is rejected, and supports the Alternate Hypothesis that the XAI techniques differ in terms of their merits in explaining credit card fraud.

Using the data in Table 4.2 as the input to a Friedman test, the following statistics were generated;

Table 4.3: Friedman Test Statistics

Metric	**Statistic**	**P-Value**	**Significant Difference**
Identity	47.76	2.40e-10	Yes
Stability	1.56	0.668493	No
Separability	55.4	5.64e-12	Yes
Similarity	60.00	5.88e-13	Yes
Computational Efficiency	60.00	5.88e-13	Yes

In the context of this Friedman test, the results indicate that there is a statistically significant difference between the four XAI techniques for the metrics *Identity*, *Seperability*, *Similarity*, and *Computational Efficiency*, as their p-values are below the threshold of 0.05. However, for *Stability*, there is no significant difference between the techniques, as the ***p-value*** is above 0.05.

The level of significance set at 0.05 is a conventional threshold in statistical testing, representing a balance between the Type I error (false positive) and the Type II error (false negative). By setting this standard, researchers accept a 5% risk of incorrectly rejecting a true NULL hypothesis, which is generally considered an acceptable trade-off to avoid more frequent or less detectable errors in most research contexts. This analysis is based on the numbers captured for the XAI metrics in the earlier experiments. These findings underscore the importance of considering multiple

evaluation metrics when assessing the effectiveness of XAI techniques, as no single technique consistently outperformed the others across all the aspects evaluated.

Graphical View of XAI Metrics Distribution - Box Plot

After statistical analysis, generating both a box plot for the XAI metric data (average scores) helps to visualise the distribution and spread of the results across the four XAI methods. Normalisation of data is not a prerequisite for the Friedman test, as this non-parametric method is designed to handle data that may not adhere to a normal distribution by comparing ranks rather than actual values. However, it is useful to perform this transformation to improve the presentation of a graphical analysis.

In the **Box Plot**, the x-axis now correctly represents the four XAI methods (SHAP, LIME, ANCHORS, and DiCE). The box shows the interquartile range (IQR), indicating the middle 50% of the scores for each method. The median is represented by the line within the box. The whiskers extend to show the range of the data, excluding outliers. This plot below allows us to compare the central tendency and variability of the scores across the different XAI methods. The *computational efficiency* score for ANCHORS will distort the visual representation of the XAI scores, so it has been removed from the Box Plot.

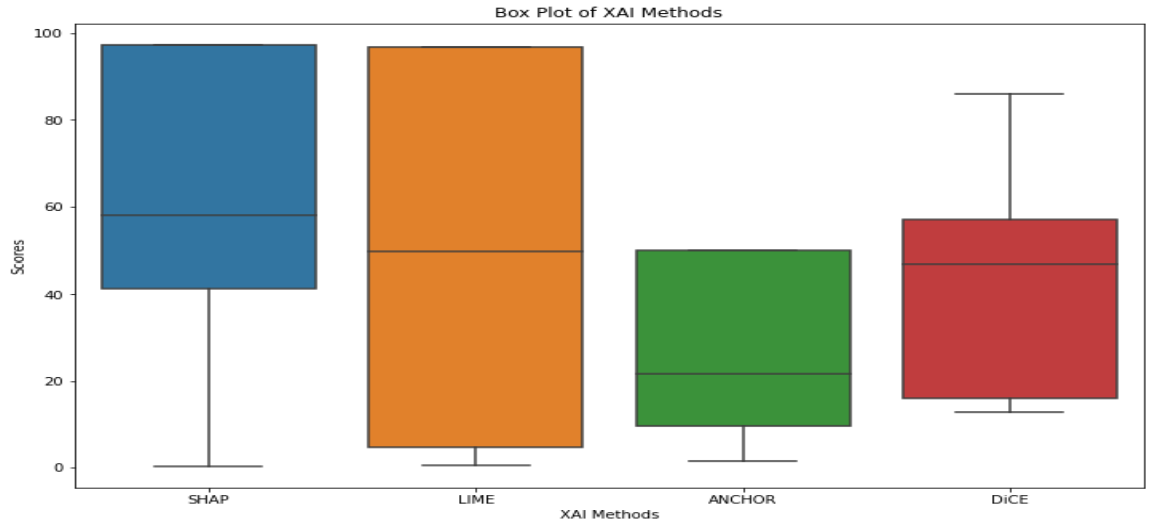


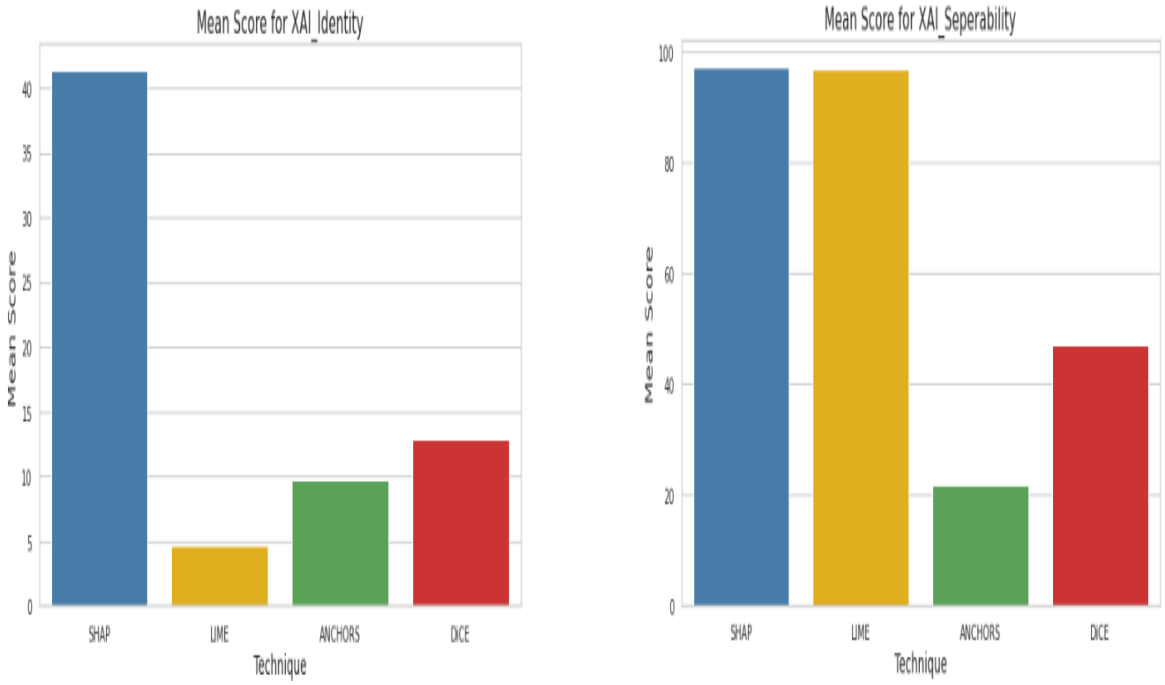
Figure 4.16: Box Plot Analysis of XAI Distributions

Graphical View of XAI Metrics Distribution - Bar Charts

The bar charts visualising the average scores of the four XAI techniques (SHAP, LIME, ANCHORS, and DiCE) across different evaluation metrics provide a clear depiction of their comparative performance.

For metrics such as *Identity* and *Separability*, significant variability is observed in mean scores between techniques, indicating that some techniques are more adept at preserving the identity of the input data and separating model predictions.

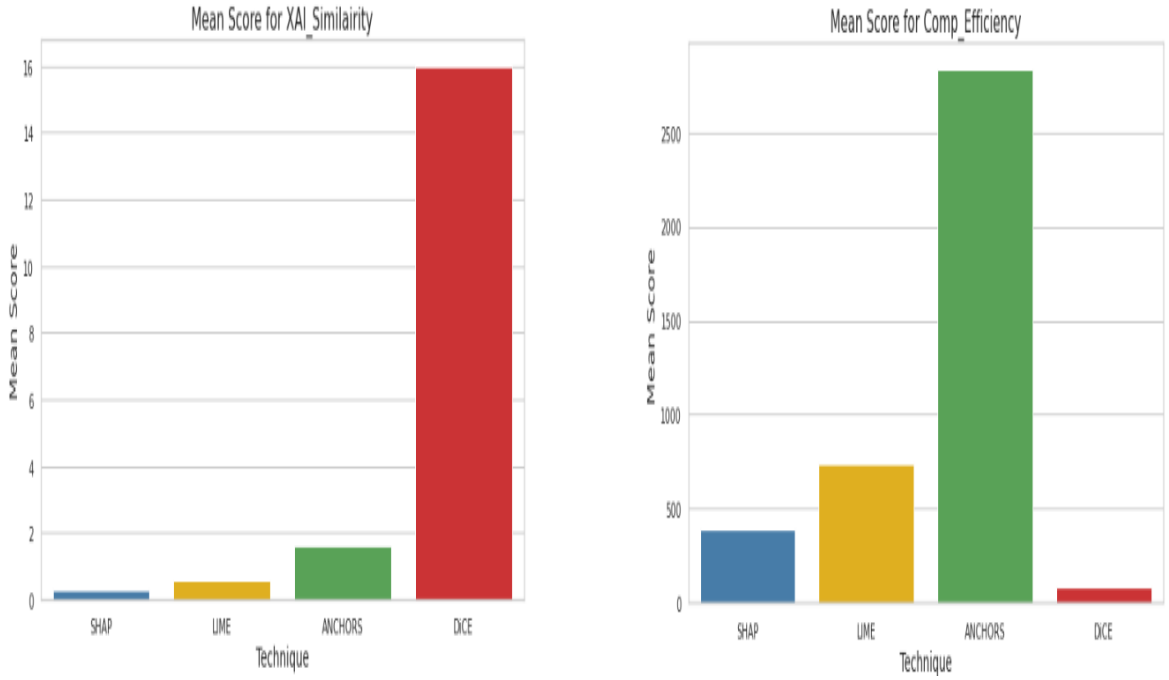
Specifically, the bar heights for SHAP and LIME reveal a marked difference in their performance on these metrics, with SHAP generally achieving higher average scores, suggesting a more consistent ability to meet these evaluation criteria. On the other hand, the closer bar heights for ANCHORS and DiCE on certain metrics ((a) and (b)) suggest similar performance levels between these two techniques in some aspects.



(a) Identity Metric Scores

(b) Seperability Metric Scores

Figure 4.17: Identity and Seperability Metric Scores Per Technique



(a) Similarity Metric Scores

(b) Comp Efficiency Metric Scores

Figure 4.18: Identity and Seperability Metric Scores Per Technique

The visualisations also highlight that no single XAI technique dominates across all metrics, underscoring the trade-offs involved in selecting an appropriate technique for specific explanatory goals. Ultimately, these bar graphs serve as a visual summary of the statistical analysis, enabling a straightforward comparison of the effectiveness of the XAI techniques across a variety of custom evaluation metrics.

4.3.2 Wilcoxon Signed-Rank Test Analysis

The Wilcoxon signed rank test is a non-parametric test used to compare two related samples to assess whether their population mean ranks differ. It is an alternative to the paired t test when it cannot be assumed that the data are normally distributed.

To attempt to determine a ranking of each XAI technique based on the custom metrics and to establish which XAI technique is *the best*, a pairwise comparison is performed between all the techniques for each metric.

Key points to note in this comparison are;

1. Multiple Comparisons Issue: Conducting multiple pairwise comparisons increases the risk of committing Type I errors (false positives). To address this, the p values are adjusted using a Bonferroni correction method to maintain the overall Type I error rate.

2. Ranking and *The Best*: Although the Wilcoxon test can identify significant differences between pairs of techniques, determining *the best* technique would require a more nuanced approach. For example, in many cases, an XAI technique that performs best in one metric might not perform as well in another. A composite score or a decision based on the weighted priorities of each custom metric may be necessary. This topic will be discussed further in the Conclusions section of this paper; Chapter 5.

3. Non-parametric Nature: The Wilcoxon test assesses differences in median ranks rather than means. Thus, *the best* in this context would be related to which technique consistently ranks higher on the evaluated metrics, not necessarily which has the highest mean scores.

Tabular View of Wilcoxon Results

	Metric	Comparison	Statistic	p-Value	Significant*
1	**Identity**	**SHAP vs LIME**	0.0	0.000002	Yes
2	**Identity**	**SHAP vs ANCHORS**	0.0	0.000002	Yes
3	**Identity**	**SHAP vs DiCE**	0.0	0.000002	Yes
4	**Identity**	**LIME vs ANCHORS**	25.0	0.001690	Yes
5	**Identity**	**LIME vs DiCE**	9.0	0.000063	Yes
6	**Identity**	**ANCHORS vs DiCE**	62.0	0.113987	No
7	**Stability**	**SHAP vs LIME**	76.0	0.294252	No
8	**Stability**	**SHAP vs ANCHORS**	81.0	0.388376	No
9	**Stability**	**SHAP vs DiCE**	105.0	1.000000	No
10	**Stability**	**LIME vs ANCHORS**	103.0	0.956329	No
11	**Stability**	**LIME vs DiCE**	62.0	0.113987	No
12	**Stability**	**ANCHORS vs DiCE**	75.0	0.277355	No
13	**Seperability**	**SHAP vs LIME**	48.5	0.508569	No
14	**Seperability**	**SHAP vs ANCHORS**	0.0	0.000002	Yes
15	**Seperability**	**SHAP vs DiCE**	0.0	0.000002	Yes
16	**Seperability**	**LIME vs ANCHORS**	0.0	0.000002	Yes
17	**Seperability**	**LIME vs DiCE**	0.0	0.000002	Yes
18	**Seperability**	**ANCHORS vs DiCE**	0.0	0.000002	Yes
19	**Similarity**	**SHAP vs LIME**	0.0	0.000002	Yes
20	**Similarity**	**SHAP vs ANCHORS**	0.0	0.000002	Yes
21	**Similarity**	**SHAP vs DiCE**	0.0	0.000002	Yes
22	**Similarity**	**LIME vs ANCHORS**	0.0	0.000002	Yes
23	**Similarity**	**LIME vs DiCE**	0.0	0.000002	Yes
24	**Similarity**	**ANCHORS vs DiCE**	0.0	0.000002	Yes
25	**Comp Efficiency**	**SHAP vs LIME**	0.0	0.000002	Yes
26	**Comp Efficiency**	**SHAP vs ANCHORS**	0.0	0.000002	Yes
27	**Comp Efficiency**	**SHAP vs DiCE**	0.0	0.000002	Yes
28	**Comp Efficiency**	**LIME vs ANCHORS**	0.0	0.000002	Yes
29	**Comp Efficiency**	**LIME vs DiCE**	0.0	0.000002	Yes
30	**Comp Efficiency**	**ANCHORS vs DiCE**	0.0	0.000002	Yes

Table 4.4: Wilcoxon Signed-Rank Pairwise Tests

*(α=0.05, Bonferroni corrected)

The full table presenting the output of the Wilcoxon signed-rank tests for pairwise comparisons between the XAI techniques for all the "XAI" metrics, with Bonferroni correction applied for multiple comparisons, is presented in Table 4.4 above.

The Wilcoxon signed-rank test results provide two key pieces of information: the 'Statistic' and the 'P-Value'.

The *'Statistic'* value in the context of the Wilcoxon test represents the sum of positive ranks in the differences between paired samples of the XAI methods. It reflects the magnitude and direction of the differences, but does not directly translate to the size of the effect or its practical significance. A lower statistic suggests more consistent differences in favour of one sample over the other within each pair of XAI methods.

The *'P-Value'* assesses the probability that the observed differences could have occurred under the NULL hypothesis (which states that there is no difference between the pairs). A low p-value (determined below 0.05 for these XAI experiments) would indicate that the differences observed between the XAI explainers are statistically significant, suggesting that one XAI method consistently differs from the other in its performance (based on the metrics used in this research paper).

The Bonferroni correction is applied in the analysis to adjust the significance threshold for each test, reducing the chance of false positives due to the increased risk of Type I errors when performing multiple pairwise comparisons, by dividing the standard alpha level (e.g., 0.05) by the number of tests conducted.

Table 4.4 demonstrates the statistical significance of the differences between the XAI techniques on different metrics, adjusted for multiple comparisons using the Bonferroni correction. In the analysis of Explainable Artificial Intelligence (XAI) techniques using custom evaluation metrics, Wilcoxon signed-rank tests with Bonferroni correction revealed statistically significant differences between pairs of techniques across various metrics, indicating variations in their explanatory performance. The *Identity* metric showed significant differences for most pairwise comparisons, except between ANCHORS and DiCE, suggesting that these two

techniques have comparable performance in preserving the identity of instances in their explanations. In contrast, the *Stability* metric displayed no significant differences among any of the XAI techniques, which implies uniform stability between the techniques when providing explanations. For *Seperability* and *Similarity*, significant differences were observed in all pairwise comparisons, highlighting notable variations in how different techniques manage to differentiate between and accurately reflect similarities of instances, respectively. These findings suggest that certain XAI techniques may be more suited to specific evaluative criteria, depending on the desired aspect of explainability, such as identity preservation or accuracy in reflecting instance similarities.

The Wilcoxon signed-rank test, as used in this analysis, can identify statistically significant differences between pairs of XAI techniques across various evaluation metrics but does not inherently provide a ranking or determine the absolute performance levels of the techniques. It highlights whether there is a significant difference in the distributions of scores for the paired comparisons but does not quantify the magnitude of difference in a way that would allow for a straightforward ranking. Moreover, since *the best* XAI technique would depend on the specific metric of interest and the Wilcoxon test only indicates if one technique outperforms another for a given metric without aggregating these outcomes into a comprehensive score, it is not possible to directly conclude which XAI technique is superior overall. Each technique's performance is context-dependent, and what is considered ***best*** may vary based on the importance or relevance of each metric to the specific application or user needs. Thus, while significant results from the Wilcoxon tests guide us in understanding comparative strengths and weaknesses, establishing a definitive ranking or the *best* technique requires further analysis, potentially involving different statistical methods or decision frameworks that account for the relative importance of each metric.

4.3.3 Visualisation of Metric Score Results

A density plot from the experiment data will consider the distribution of scores between all combined methods and metrics. This graph aggregates all scores into a single distribution, allowing us to create a comprehensive density plot.

To reduce the effect of the spread of values for the ANCHOR method in a density graph, a logarithmic transformation is applied to all values to reduce the skewness and bring the distributions closer together.

This transformation in Figure 4.19 makes it easier to observe the distributions of each method on a similar scale, especially since the ANCHOR method had *Comp_Efficiency* values that were significantly higher than those of other XAI methods.

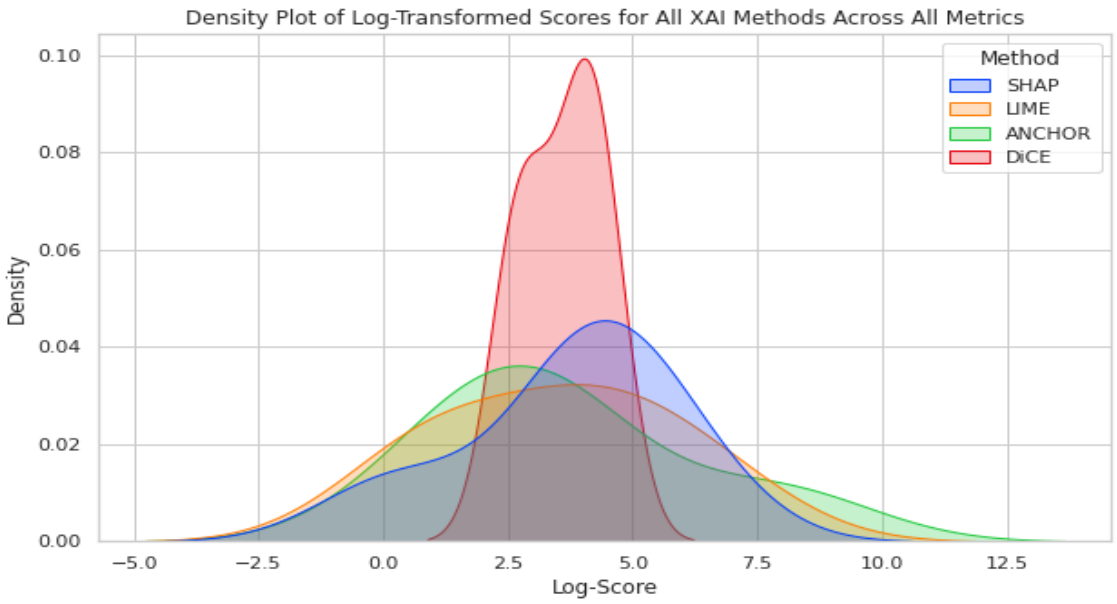


Figure 4.19: Density Plot of Log-Transformed Scores for All XAI Methods Across All Metrics

In these density plots, the distributions of scores for our four Explainable Artificial Intelligence (XAI) methods - SHAP, LIME, ANCHOR, and DiCE - are visualised across five custom metrics, providing insight into their performance characteristics. In particular, the ANCHOR method exhibits a distinctively broad spread of values,

indicating high variability in its performance across the evaluated metrics, a pattern that is mitigated through logarithmic transformation to enable better comparability. The SHAP and LIME methods show slightly more concentrated distributions, suggesting a more consistent performance across the metrics. The application of these density plots highlights the varying degrees of effectiveness and reliability of each XAI method in different evaluative contexts, underlining the importance of method selection based on specific metric requirements. This comparative analysis is crucial for data scientists in choosing the most suitable XAI method for a given application, ensuring an optimal balance between interpretability, accuracy, and reliability.

4.3.4 Assessment of Experiment Results

Sections 4.3.1 and 4.3.2 provide clear statistical evidence that none of the chosen XAI explainers clearly outperforms the others.

A similar set of experiments conducted by (ElShawi et al., 2020) also came to a comparable conclusion, stating that after their experiments the *"..results show that there is no clear winner..."*. The datasets in that research were focused on healthcare classification problems and the performance of the XAI methods against their metrics was presented in a similar tabular format. No statistical tests were performed for the difference.
In this research, the domain under scrutiny is the classification of credit card fraud. The logic behind the XAI metrics themselves is similar to the ElShawi et al. approach, but the assessment of performance is more statistically rigorous. Despite these differences, the end result is the same. Once the performance of the XAI explainers is reduced to a statistical analysis, it is possible to determine the relative merits or drawbacks of each technique, but it is not possible to declare that one method is demonstrably *better*.

Chapter 5

Conclusion

5.1 Summary

The Search for a Statistically Significant Ranking of XAI Methods

This research sought to examine the challenge of proposing a quantifiable framework to assess Explainable Artificial Intelligence methods (XAI) that provide a reason why a given credit card transaction is labelled as fraudulent. The intention was to provide an objective analysis as to whether a given state-of-the-art ML explainer could be shown to provide the best explanations, when compared against other explainer techniques in this particular financial crime domain. The experiments in this thesis showed that, after training a Neural Network model on a credit card fraud dataset, it **is** possible to distinguish between the merits of the SHAP, LIME, ANCHORS, and DiCE interpretability methods used in the experiments in this research. The XAI techniques scored differently on four of the five custom metrics. An observer could declare one technique to be the *best*, but only if a weighting was applied to one or more of the metric scores. No such weightings were applied in the experiments in this thesis research.

Taking into account recent research on the value of XAI techniques, it was found that it is common to use human-led surveys and assessments to interpret the value of explainer results. In a given domain, both experts and non-experts are asked to rate

how effective they felt the explainer outputs were in terms of providing a meaningful explanation. This research paper took a deliberate approach to avoid such an experiment structure, which can be expensive to implement and difficult to replicate effectively over successive time periods. A common set of metrics was defined, and the explainer outputs for the SHAP, LIME, ANCHORS, and DiCE techniques were *scored*.

The preference for this type of automated experimental evaluation in this research question emerged during the literature review phase of this dissertation. However, that focus was on the area of various healthcare classification problems. By moving the domain to the area of credit card fraud detection, the experiments in this thesis sought to determine whether one of the four chosen techniques could be shown to be demonstrably statistically stronger in explaining the causes of an individual classification of *fraud*.

Key Observations/Conclusions from the Mean Metrics Scores Across the XAI Methods

- The *Stability* score is the only metric with consistent values across the XAI Methods. On average, 50% of the explanations clusters match the grouping of fraud and non-fraud instances. Variations are more clear-cut across the other metrics.

- *Identity* scores poorly for all XAI methods, with the exception of SHAP. This was arguably the most straightforward metric: similar instances should have similar explanations. However, the SHAP technique provides a score for *all* instance values, while the other methods generate explainers that cover only *some* instance attributes. This does not invalidate the use of this metric, but it is a key consideration when evaluating this research.

- *ANCHORS* is one of the methods that produces relatively sparse output (only one feature might be classified as an 'anchor' in the classification result). Thus, the explanations score relatively poorly in distinguishing themselves from each other (*Seperability*).

- The magnitude of counterfactual explanations for different individual instances can vary significantly. Hence, the DiCE method produces outputs in which the Euclidean distance between separate explanations can be significant. This characteristic explains the higher DiCE score for *Similarity*.

- The *Computation Efficiency* score may seem out of place in this thesis, as it may be considered a measure of general performance as opposed to a metric of the quality of the explainer output. However, the speed (or lack thereof) with which an explainer can process this credit card dataset is deemed a critical measure for comparison. DiCE and SHAP generation time is markedly quicker than ANCHORS for example. Even with parallel/scaleable batch processing options, the user of Anchors for explanations in a commercial high-volume fraud detection environment may not be viable.

- All custom metrics were designed and implemented to score a range of traits across the XAI methods. A poor score on one metric is often balanced by better performance on other metrics. The research appears to have applied a balanced assessment framework for use in these experiments. The statistical analysis of these experiment outputs did not rank a clear *winner*. However, if a user were to value the *Similarity* and *Computational Efficiency* scores over the other metrics, then this would penalise Anchors and DiCE and strongly favour SHAP as the preferred technique.

Validity of the Statistical Analysis of the XAI Techniques?

The experiments carried out in this research established that repeatable statistical analysis was possible and that comparisons could be drawn between different XAI techniques.
During the execution of the XAI metrics experiments and the analysis of the results, the following observations emerged about this type of statistical analysis:

- The output from LIME, Anchors, and DiCE is immediately understandable by a human reader. It could be argued that this is the strength of their output

and that these techniques have not been built for statistical analysis. However, tailoring this type of interpretability output for a statistical comparison analysis is one of the stated objectives of this research.

- The DiCE algorithm was ineffective in producing counterfactual explanations until an increased number of continuous features, with a wider range of values, were added to the model creation process. Thus, when building the credit card fraud predictive model, it was necessary to include extra columns from the original dataset that provided additional information on the transaction amount, such as the size (in dollars) of other transactions on the same card that occurred immediately before or after the given transaction. The addition of this data produced interesting explanations and allowed the DiCE XAI metrics to be generated for the thesis experiments, but these newer features had an almost imperceptible impact on model performance. This highlights an interesting comparison with the research conducted by (ElShawi et al., 2020) on interpretability techniques for healthcare datasets. The data used in those 2020 experiments were built from treadmill exercise stress tests and contained 43 numerical attributes such as age, resting systolic blood pressure, obesity scores, etc. Many of these types of feature obviously provided a sufficient range of values in the dataset that would have fed effectively into a counterfactual XAI algorithm, and were also pertinent in model performance. In the credit card dataset used for this dataset, the predictive model was strongly influenced by binary attributes, such as the use (or not) of a PIN (*PinIndicator*), was a cashier present at the POS terminal (*PosTerminalAttended*), was the customer present at the transaction (*CustomerPresent Indicator*), or was this an E-Commerce transaction (*ECommerceFlag*). Other features such as the 'Authorisation Response Recorded' (*AuthResponse*) had relatively few unique values, but were also highly relevant to the model's fraud classification predictions. Intuitively, these types of data element would be important in any ML classification problem for the detection of credit card fraud. However, a feature set with a high ratio of this type of attributes will struggle to generate

meaningful counterfactuals, based on the results seen during the experiments described in Section 4.2.4 of this thesis. The necessity to "pad" the feature list with additional continuous predictors when building a credit card fraud model raises questions around the suitability of an XAI technique such as DiCE counterfactuals in future experiments.

- The computational overhead to generate the ANCHOR explanations almost derailed the execution of the experiment itself. It took 18+ hours, running at intervals over three days, to complete the output show in Section 4.2.3 on the available infrastructure. Adding checks and output displays to the Python Notebook showed that approximately 95% of the processing time was being consumed by just less than 5% of the test data instances. Despite tuning the XAI Anchor function itself, it was never possible to impact on this ratio. For Anchor explanations, it could be theorised that additional preprocessing of the credit card dataset, or a refinement of the *Anchor_Tabular* algorithm, might mitigate the characteristics that cause increased complexity for specific instance rows. Generating Anchor inputs for the XAI metrics functions was an extremely time-consuming process on the available hardware. Options to streamline the Anchor interpretations would be of great benefit for future iterations of these experiments. However, such an analysis is beyond the current scope of this thesis.

The conclusion drawn from the above points, which arise from observations on the execution and results of the experiments in this thesis, is that the researcher must be aware of **how** the XAI technique manages the characteristics of the source data. Statistical analysis is a useful tool and there are insights to be gained, but the value of the metric scores must be considered in conjunction with how the individual experiment handled the source data.

We have seen that the credit card dataset in this thesis has issues in generating explanations through the ANCHOR and DiCE processes. The initial setup of the Python Notebooks experiments in this research used a relatively simple Kaggle

credit risk dataset (https://www.kaggle.com/datasets/laotse/credit-risk-dataset). This was not a dataset related to fraud, but it is in the Financial Services domain. The XAI methods were able to generate explanations for predicted 'default' classifications without any additional complication.
The credit card fraud dataset itself contained a number of attributes with a simple *Yes/No* flag. This characteristic was not handled well by the DiCE algorithm, and additional transaction attributes had to be added to the model-build process. This transactional data provided useful information on how a classification would change if a value was altered, but such an explanation could also obscure a more important influence on the prediction score. For example, it would be important to know that if the dollar amount of the last daily transaction on the card exceeded a certain threshold, then the classification would switch to *fraud*, but that explanation may obscure the more important fact that a cashier was not present at the Point of Sale (y/n).
Although it was eventually possible to generate an ANCHORS output for the XAI metrics experiments, the long processing time involved significantly impacted the *Computational Efficiency* scores for this method. In practice, this would possibly disqualify ANCHORS from use in a commercial fraud detection investigation system. If future research established a means to address this problem on a dataset like the one used in this thesis, then that might warrant the reinclusion of ANCHORS.

Thus, the conclusion tends to support that this analysis does provide insight, but the interaction of data and interpretability technique have to be carefully considered.

5.2 Contributions and Impact

In these experiments, it could not be proven that one of the chosen XAI techniques (SHAP, LIME, ANCHORS, or DiCE) would provide superior explanations for an NN model that identified a particular credit card transaction instance as fraudulent. SHAP values are already a relatively common set of data points that commercial products in the Financial Crime space use when they offer local *post-hoc* card fraud

explanations to users. One might infer from the experimental output of the research in this paper that there may be moderate benefits for vendors to develop additional interfaces that provide LIME, ANCHORS, or DiCE style outputs, depending on the relative merits assigned to the custom metrics. However (and obviously), this paper should only be considered as a starting point for future product roadmap research/development. The credit card dataset used in this paper contained many useful fraud patterns, but is only a single source.

The XAI techniques chosen for the experiments in this paper are commonly referenced and used in the interpretability research found in the literary review in this thesis. However, there are other methods and variations on SHAP, LIME, ANCHORS, and Counterfactuals for which their output could be scored by the *Identity*, *Stability*, *Separability*, *Similarity*, and Computational Efficiency metrics designed for this article.

5.3 Future Work

Broaden Range of XAI Techniques

An objective and repeatable framework to assess the quantifiable benefits of XAI techniques for credit card (or other) fraud is clearly attractive for Product Managers who constantly wish to iterate on new offerings in this financial crime prevention marketplace. The experiments in this paper were limited to four XAI techniques, but fraud classification would also be suitable for interpretability processes such as **LORE**, **ILIME**, **MAPLE**, and others. An obvious evolution of the experiments in this paper is to extend the breath of explainers and increase the matrix of metrics that are input into the Friedman/Wilcoxon-Paired tests. Repeating experiments on multiple credit card fraud datasets will also increase the variety of data elements considered by the explainer processes.

The choice of future XAI methods is particularly relevant because the experiments in this thesis have shown that the combination of dataset characteristics and XAI processes can potentially undermine any purely statistical analysis.

Recommendations

The rapid advancement of Generative AI in 2023/2024 is having a striking impact on commercial product development and individual productivity alike. XAI techniques have been helping, in recent years, to unlock the *black-block* decisions made by Neural Network models in financial decisions. However, the advent of LLM plug-ins in commercial products will provide an interface that can provide a 'human-like' narrative to explain a classification to the end user. A fascinating progression to the research in this paper would be to feed in multiple XAI outputs into a portal such as OpenAI and then allow that model to build a comprehensive text explanation based on the features of each technique. It may be less a question of which interpretability process is best in local *post hoc* classification for credit card fraud, and more a case of combining all these XAI explanations into a GenAI human readable description. The experiments in this paper would imply that the processing power required to generate multiple local explanations in a commercial system might be prohibitive for anything other than parallel batch runs, but this may just be a short-term limitation.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, *6*(52), 138–160. doi: 10.1109/access.2018.2870052

Ajitha, E., Sneha, S., Makesh, S., & Jaspin, K. (2023). A comparative analysis of credit card fraud detection with machine learning algorithms and convolutional neural network. In *2023 international conference on advances in computing, communication and applied informatics (accai)* (p. 1-8). doi: 10.1109/ACCAI58221.2023.10200905

Alvarez-Melis, D., & Jaakkola, T. (2018). On the robustness of interpretability methods. *2018 ICML Workshop on Human Interpretability in Machine Learning*. doi: 10.48550/arXiv.1806.08049

Anowar, F., & Sadaoui, S. (2020). Incremental neural-network learning for big fraud data. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, *1*(1), 1–4. doi: 10.1109/smc42975.2020.9283136

Aurna, N. F., Hossain, M. D., Taenaka, Y., & Kadobayashi, Y. (2023). Federated learning-based credit card fraud detection: Performance analysis with sampling methods and deep learning algorithms. In *2023 ieee international conference on cyber security and resilience (csr)* (p. 180-186). doi: 10.1109/CSR57506.2023.10224978

Batageri, A., & Kumar, S. (2021). Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, *2*(1), 35–41. doi: 10.1016/j.gltp.2021.01.006

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019, Jul). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8). doi: 10.3390/electronics8080832

Chromik, M., .Eiband, M., Buchner, F., Krüger, A., & Butz, A. (2021). I think i get your point, ai! the illusion of explanatory depth in explainable ai. *26th International Conference on Intelligent User Interfaces*. doi: 10.1145/3397481.3450644

Dal Pozzolo, A., et al. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, *41*(10), 4915–4928. doi: 10.1016/j.eswa.2014.02.026

Darias, J. M., Caro-Martínez, M., Díaz-Agudo, B., & Recio-Garcia, J. A. (2022, Aug). Using case-based reasoning for capturing expert knowledge on explanation methods. *Case-Based Reasoning Research and Development*, *13405*, 3–17. doi: 10.1007/978-3-031-14923-8_1

ElShawi, R., Sherif, Y., Al-Mallah, M., & Sakr, S. (2020, Aug). Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. *Computational Intelligence*, *37*(4), 1633–1650. doi: 10.1111/coin.12410

Evans, B. P., Xue, B., & Zhang, M. (2019, Jul). What's inside the black-box? *Proceedings of the Genetic and Evolutionary Computation Conference*. doi: 10.1145/3321707.3321726

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., & Turini, F. (2019, Dec). Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, *34*(6), 14–23. doi: 10.1109/mis.2019.2957223

Hanafy, M., & Ming, R. (2022). Classification of the insureds using integrated machine learning algorithms: A comparative study. *Applied Artificial Intelligence*, *36*. doi: 10.1080/08839514.2021.2020489

Honegger, M. (2018, Aug). *Shedding light on black box machine learning algorithms: Development of an axiomatic framework to assess the quality of methods that explain individual predictions.* Karlsruhe Institute of Technology. Retrieved from `https://arxiv.org/abs/1808.05054v1`

Ignatiev, A. (2020, Jul). Towards trustable explainable ai. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 5154–5158. doi: 10.24963/ijcai.2020/726

Jacob, V., Song, F., Stiegler, A., Rad, B., Diao, Y., & Tatbul, N. (2021). Exathlon: A benchmark for explainable anomaly detection over time series. *Proceedings of the VLDB Endowment*, *14*(11), 2613–2626. doi: 10.14778/3476249.3476307

Jesus, S., Belém, C., Balayan, V., Bento, J., Saleiro, P., Bizarro, P., & Gama, J. (2021, Mar). How can i choose an explainer? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. doi: 10.1145/3442188.3445941

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., & Wortman-Vaughan, J. (2020, Apr). Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. doi: 10.1145/3313831.3376219

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016, Aug). Interpretable decision sets. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1675–1684. doi: 10.1145/2939672.2939874

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems 30 (nips 2017)* (Vol. 30). NeurIPS Proceedings.

Marcilio, W. E., & Eler, D. M. (2020, Nov). From explanations to feature selection: Assessing shap values as feature selection mechanism. *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 340–347. doi: 10.1109/sibgrapi51738.2020.00053

Martínez, M., Nadj, M., Langner, M., Toreini, P., & Maedche, A. (2023). Does this explanation help? designing local model-agnostic explanation representations and an experimental evaluation using eye-tracking technology. *ACM Transactions on Interactive Intelligent Systems*. doi: 110.1145/3607145

Moreira, C., Chou, Y., Velmurugan, M., Ouyang, C., Sindhgatta, R., & Bruza, P. (2020). Linda-bn: An interpretable probabilistic approach for demystifying black-box predictive models. *Decision Support Systems*, *150*. doi: 10.1016/j.dss.2021.113561

Mothilal, S. A., R. K., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. doi: 10.48550/arXiv.1806.08049

Nascita, A., Montieri, A., G.Aceto, Ciuonzo, D., Persico, V., & Pescape, A. (2021). Unveiling mimetic: Interpreting deep learning traffic classifiers via xai techniques. *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 455–460. doi: 10.1109/csr51186.2021.9527948

Nesvijevskaia, A., Ouillade, S., Guilmin, P., & Zucker, J. (2021). The accuracy versus interpretability trade-off in fraud detection model. *Data and Policy*, *3*. doi: 10.1017/dap.2021.3

Nguyen, M., Bouaziz, A., Valdes, V., Rosa-Cavalli, A., Mallouli, W., & MontesDeOca, E. (2023). A deep learning anomaly detection framework with explainability and robustness. *Proceedings of the 18th International Conference on Availability, Reliability and Security.*. doi: 10.1145/3600160.3605052

Nri, H., Jenkins, S., Paul, K., & Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Priscilla, C., & Prabha, D. (2020). Influence of optimizing xgboost to handle class imbalance in credit card fraud detection. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 1309–1315. doi: 10.1109/icssit48917.2020.9214206

Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. (2021). Explainable machine learning for fraud detection. *Computer*, *54*(10), 49–59. doi: 10.1109/mc.2021.3081249

Ras, G., Xie, N., Gerven, M. V., & Doran, D. (2022). Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, *73*, 329–397. doi: 10.1613/jair.1.13200

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, Aug). "why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. doi: 10.1145/2939672.2939778

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018, Feb). Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). doi: 10.1609/aaai.v32i1.11491

Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., & Keim, D. (2019). Towards a rigorous evaluation of xai methods on time series. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*.. doi: 10.1109/iccvw.2019.00516

Sharma, A., & Bathla, N. (2020, Aug).

Review on credit card fraud detection and classification by Machine Learning and Data Mining approaches, *6*(4), 687–692.

Sharma, P., & Priyanka, S. (2020, Jun). Credit card fraud detection using deep learning based on neural network and auto encoder. *International Journal of Engineering and Advanced Technology*, *9*(5), 1140–1143. doi: 10.35940/ijeat.e9934.069520

Sinanc, D., Demirezen, U., & Sağıroğlu, (2021). Explainable credit card fraud detection with image conversion. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *10*(1), 63–76. doi: 10.14201/adcaij20211016376

Sullivan, R., & Longo, L. (2023). Explaining deep q-learning experience replay with shapley additive explanations. *Machine Learning and Knowledge Extraction*, *5*(4), 1433–1455. doi: 10.48550/arXiv.1806.08049

T.Y.Wu, & Y.T.Wang. (2021). Locally interpretable one-class anomaly detection for credit card fraud detection. *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. doi: 10.1109/taai54685.2021.00014

Vilone, G., & Longo, L. (2021a, May). Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, *76*, 89–106. doi: 10.1016/j.inffus.2021.05.009

Vilone, G., & Longo, L. (2021b). A quantitative evaluation of global, rule-based explanations of post-hoc, model agnostic methods. *Frontiers in Artificial Intelligence*, *4*. doi: 10.3389/frai.2021.717899

Vouros, G. (2022). Explainable deep reinforcement learning: State of the art and challenges. *ACM Computing Surveys*, *55*(5), 1–39. doi: 10.1145/3527448

Appendix A

Data Availability Statement

The data and code for this research is openly available in a public repository - https://github.com/JackDaedalus/Dissertation.git

Appendix B

Credit Card Fraud Dataset: Key Characteristics

To better understand the credit card dataset upon which the XAI methods are generating outcome explanations in the experiments in this thesis, the following visualisations were generated.

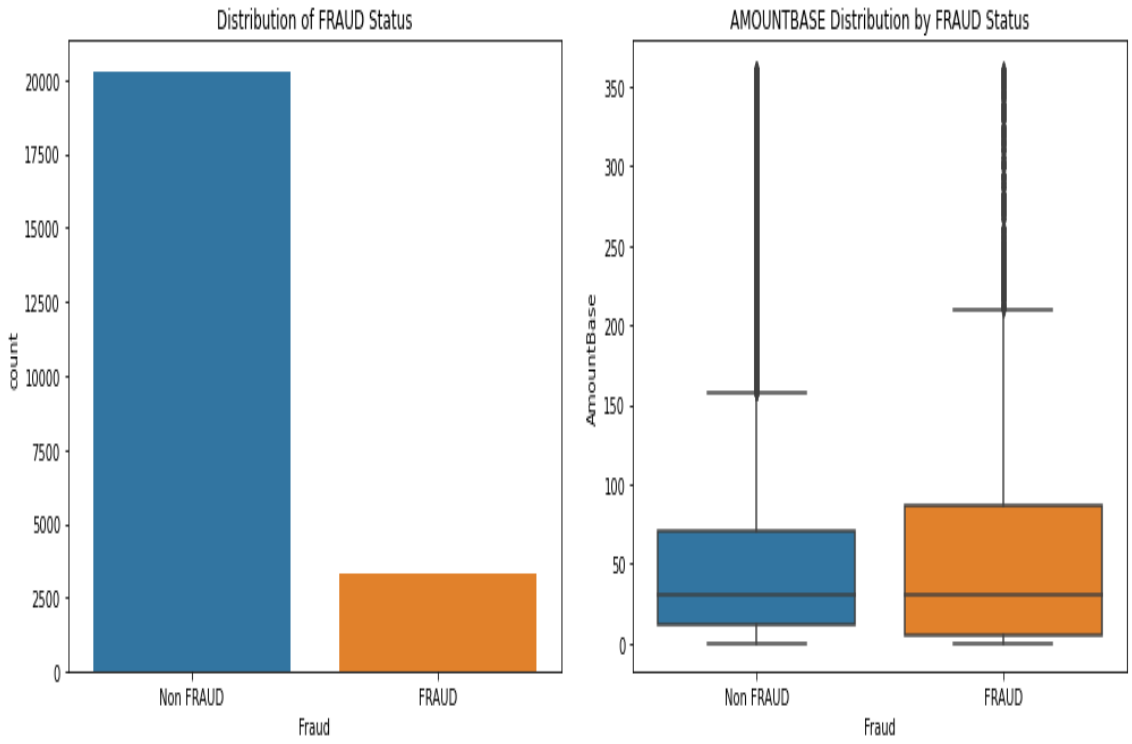


Figure B.1: Fraud Distribution and Transaction Amounts in Credit Card Dataset

The credit card fraud dataset used in these experiments is not as highly imbalanced as is common with many similar research data examples. This allows the data to be balanced, with sufficient volume, in advance of model creation without the risk of a highly dominant label distorting the model.

The bulk of the transaction amounts, for both fraud and non-fraud transactions, are less than $100. Although there is a significant volume of 'outlier' transactions in the range above $250, none are greater than $350. It should be noted that a very small number of transactions in the high thousands were removed in the feature engineering process to avoid distortion in the model-building process.

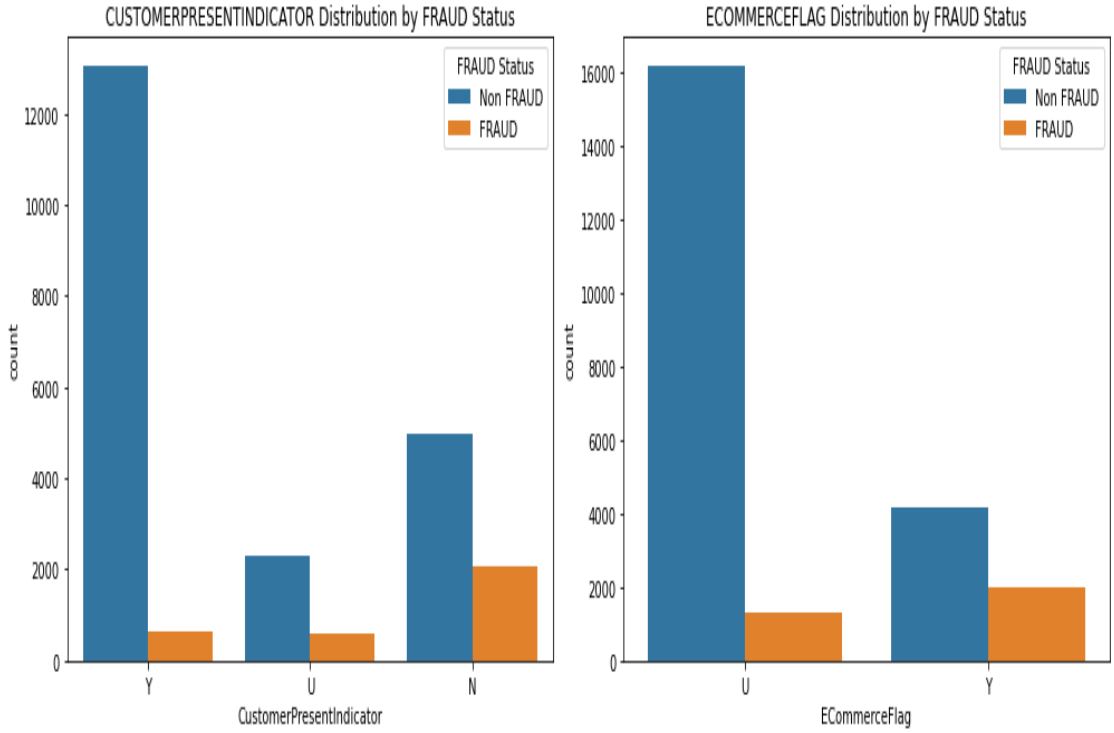


Figure B.2: Customer Present and ECommerce Flag Indicators

Fraud is much less common for credit card transactions where the buyer is present for the transactions. In contrast, the occurrence of fraud is proportionally more likely when the buyer is not physically conducting the transaction with the merchant.

As a corollary to the last data observation, E-Commerce transactions are seen to be proportionally more susceptible to fraud.

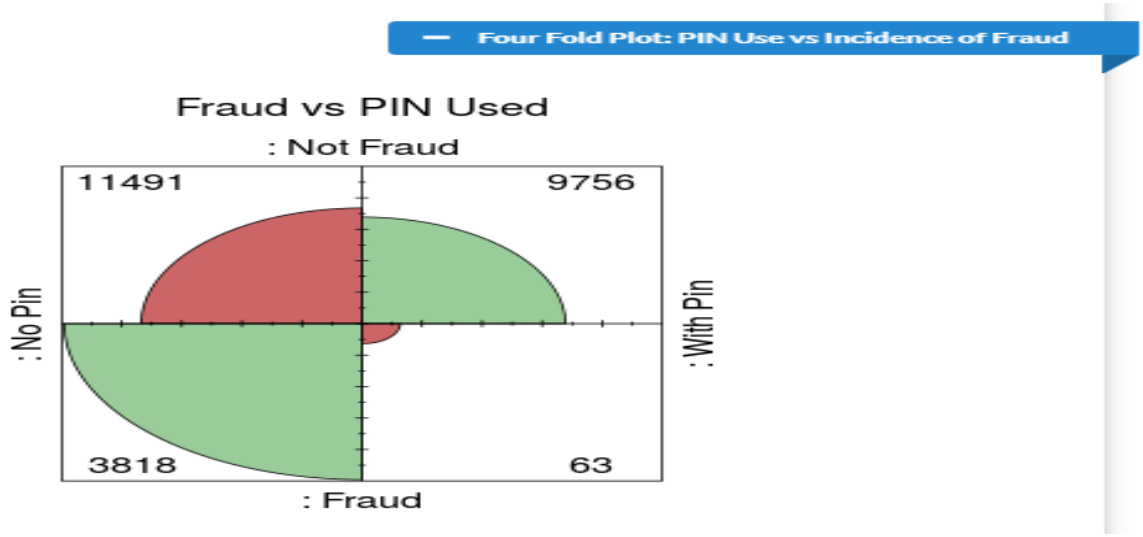


Figure B.3: PIN Use vs Incidence of Fraud

As might be expected, the incidence of fraud in which the buyer has been validated with a pin is negligible.

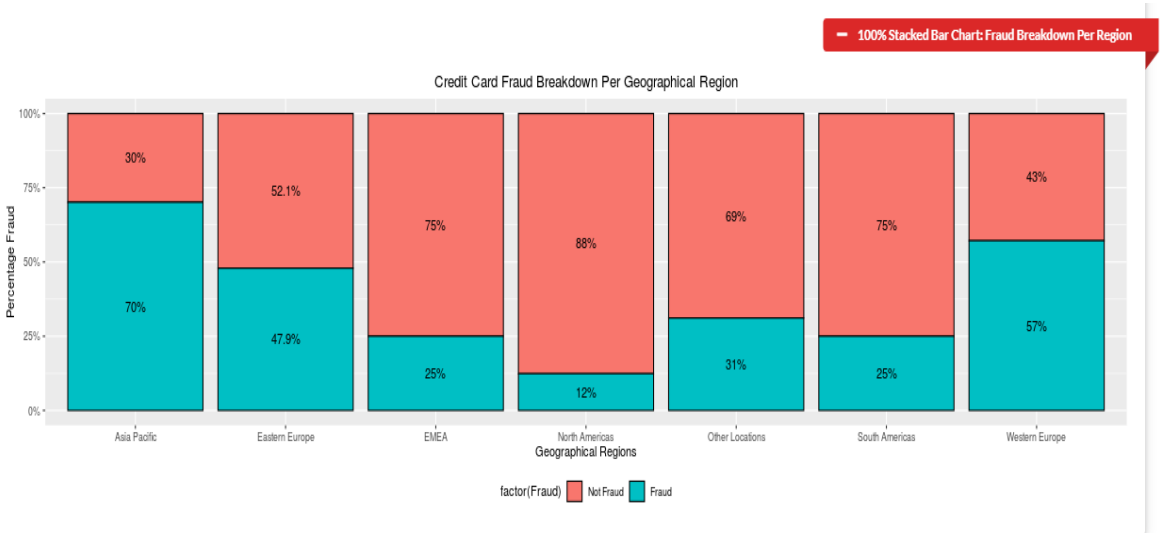


Figure B.4: CC Fraud Breakdown Per Geographical Region

The vast majority of records in the dataset were related to transactions in the US, although an element of *foreign* transaction are present. Non-US transactions show a higher incidence of fraud, but this is a smaller book of business.

Appendix C

XAI Metrics: Implementation Pseudo-code

The source code used in model building and XAI experimentation in this thesis is freely available - see Appendix A. However, the pseudo-code used to implement the Python functions for generating each metric is also provided in the following.

1. Identity

IDENTITY

Pseudocode

- Start with first instance in test data
- Search all other instances in test data and calculate distance from first instance (feature distance)
- Select closest other instance to first instance, i
- Generate explanations for all instances in test data
- Calculate distance of first instance explanations from explanations in all other instances
- Select closest other instance to first instance (explanation distance), t
- Generate success if instance id (i) = instance id (t)
- Drop first instance from test data

2. Stability

STABILITY

Pseudocode

"Stability" - this metric states that instances belonging to the same class must have comparable explanations

- Assume that the dataset has been balanced 50:50 for fraud/non-fraud.
- Cluster explanations of all instances in test data by k-means, include the 'predicted fraud' label.
- Number of clusters equals label values, in this case two (fraud/non-fraud)
- For each instance in test data
  - compare explanation cluster label to predicted class label
  - if match, then stability satisfied

  alternatively
  - compare explanation cluster label in largest cluster to predicted class label
  - Take ratio of majority predicted class label to minority class as the stability measure (the higher the value the closer the explanation clusters map to predicted results).

Question: how do we know which explanations cluster equates to 'fraud' and which cluster equates to 'non-fraud'? If dataset is a 50:50 label split and we use two clusters then we can just pick one cluster (use the largest).

The majority class in the Test data will be non-Fraud, so assume that is always the largest cluster.

3. Seperability

SEPERABILITY

Pseudocode

"Seperability" - two dissimilar instances muat have dissimilar explanations

Take subset of test data and determine for each individual instance the number of duplicate explanations in entire subset, if any.

To measure the separability metric, we choose a subset S of the testing data set that has no duplicates and get their explanations. Then for every instance s in S, we compare its explanation with all other explanations of instances in S and if such explanation has no duplicate then it satisfies the separability metric.

- Choose subset S of test data

```
- ensure no duplicate instances exist. This is a comparison of features,
  as no explanations have been generated yet.
- remove any instances with duplicated features
- generate explanations for each remaining instance in the subset of test data
```

- For every instance in S

```
- compare explanations with all other instance explanations
- if no duplicates are found; mark instance as 'success'
```

4. Similarity

SIMILARITY

Pseudocode

This metric states that the more similar the instances to be explained, the closer their explanations should be and vice versa.

To measure the similarity metric, we cluster instances in the testing data set, after normalization using DBSCAN algorithm. For each framework, we normalize the explanations and calculate the mean pairwise Euclidean distances between explanations of testing instances in the same cluster. The framework with the smallest mean pairwise Euclidean distances across its clusters is the best reflecting the similarity metric.

- Pass instances and their respective explanations to a function
- Normalise instances in the test data(DBSCAN)
- Cluster instances in test data into clusters (Note:- not just two clusters, could be more)
- Group the explanations based on the cluster to which their associated instance has been assigned
- Calculate mean pairwise Euclidean distance between explanations in each of the groups (Note:- not just two groups, could be more)
- Calculate the average of the two distance values just generated

The *Computational Efficiency* metric is captured using a custom Python decorator function. This high-order function is wrapped around each of the functions in the Kubeflow Notebooks that generate the XAI output for each technique. The time is captured in seconds, and the value is returned to the main body of the Python Notebook that executes the specific XAI experiments.