Explainable Transformers for Credit Risk

Harry Lennox

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2024

Abstract

Machine learning credit risk models play an important role in banking, but despite the widespread success of deep learning in other domains, tree-based models are still state-of-the-art for tabular data. Additionally, there is difficulty applying complex, black-box models to financial tasks like loan decisions as customers and regulators expect transparency which is not present in many complex classifiers. In this dissertation, I introduce XGFT-Transformer, a novel hybrid model capable of achieving SOTA AUC and KS performance on two tabular credit risk datasets, as well as Global Counterfactual Importance, a novel Explainable AI (xAI) algorithm capable of producing high-quality, model-agnostic local and global feature importance scores faster than current popular methods. My results demonstrate great promise for other Transformer-based credit risk architectures, and a powerful, efficient use for counterfactual explanations that avoids their usual pitfalls.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Harry Lennox*)

Acknowledgements

First and foremost, I would like to thank my supervisor, Fengxiang He, for his support and guidance during the entire planning and research period. Being able to discuss ideas, issues and advances at any time greatly helped me, and pushed me to achieve goals I originally thought I would struggle with. I would also like to thank my family and friends for always supporting and encouraging me.

Table of Contents

Chapter 1

Introduction

One of the most fundamental practices of banks and other financial institutions is lending money. The flow of cash through saving and lending is a critical service to both individuals and businesses, and one that has been of great importance to the economy on a national and global scale for centuries. Banks profit through paying interest to account holders and charging a higher rate of interest to those who apply for loans, and so long as they are capable of providing cash for any customer on request, that same money can be used by the bank to allow for borrowing of money by other customers. This gives banks massive stores of credit to use, but also creates a degree of risk - if large sums of money are borrowed and for whatever reason cannot be paid back, the bank runs the risk of major losses, or even bankruptcy. High levels of risk, in other words lending money with few restrictions, can have catastrophic consequences such as the 2008 financial crisis where massive government bailouts were required to stabilise the economy. The FCIC, reporting on the causes of the crisis, labelled ‘excessive borrowing, risky investments, and lack of transparency’ as one of several causes [18]. Therefore, the importance of Credit Risk Management - the process of deciding whether to approve a loan, cannot be understated.

1.1 Motivation

In the years since the 2008 crisis, banks were forced to rethink their strategies for credit risk. At the same time, the field of machine learning began to enter a period of rapid development with the rise of larger, more powerful learning models capable of forming a deeper understanding of the relationships between variables across massive data sources. The appeal of these high-performing models is clear, but unlike in other application areas a major issue arises when applying complex learning models to the domain of credit risk management, as well as other high-impact decision areas. These models are black-boxes - their decisions cannot be explained in the same way traditional ML models can. Deep neural networks and modern transformer models rely on billions of parameters, too much to be fully understood even by machine learning experts. A well-known example comes from OpenAI's GPT-3 in 2020, which used roughly 175 billion parameters [10]. There is a clear balance between a model's effectiveness and how intuitive and human-understandable it is, but finance is an area that requires both high performance and informed decisions. Modern banks are highly regulated, and these regulations are strongly enforced with both customers and regulators expecting transparency and clear rules when making loan decisions. Recent initiatives like the EU Artificial Intelligence Act demonstrate a clear push towards safe, trustworthy AI models [43]. The financial sector requires the ability to harness more complex, powerful models whilst still being able to explain the reasons for a decision being made, and this is where Explainable AI has become a key component.

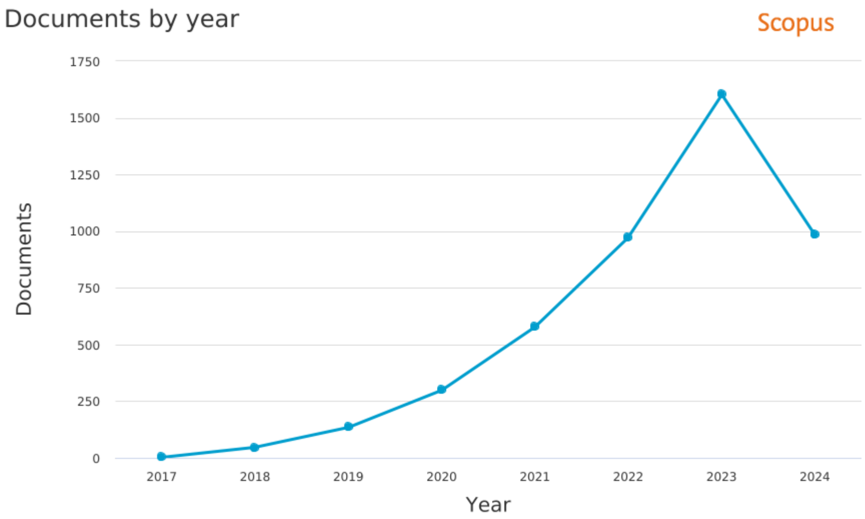


Figure 1.1: Number of Explainable AI (xAI) academic publications by year as of July 2024. (Data obtained from SCOPUS)

Explainable Artificial Intelligence or xAI is a field that is currently seeing tremendous growth as AI technology becomes incorporated into more aspects of daily life and the general public become more aware of its potential, with more papers on the subject being published every year since 2017 when this growth began. It is a wide-spanning area of research and development with many subdivisions founded upon the idea that an artificial agent must be reliable and understandable. By developing techniques to analyse how black-box models make use of their inputs to produce an observed output, we can better understand the decision process and obtain a sense of transparency. In the field of credit risk management, this technology can answer the key question of why a model approved or declined a loan, providing reliable evidence equivalent to a human bank employee's explanation that meets customer expectations and financial regulations.

Transformer-based models have recently obtained desirable, state-of-the-art results for credit risk [64] [69] [41], but they lack explainability. Explainable Transformers are currently an important area of research, but many papers on the subject focus on visual explanations or natural language applications which are unsuited to tabular data. By combining the predictive power of modified Transformers for credit risk data with efficient, informative explainability techniques, the resulting model would be well-suited for real-world explainable credit risk operation and improve the field for both banks and customers.

1.2 Contribution

In summary, the contributions of this paper are:

- A comprehensive literature review of the fields of machine learning for credit risk and explainable AI.
- A novel approach to credit risk modelling on two real-world datasets capable of outperforming the current state-of-the-art using a custom Transformer-based approach.
- A novel, model-agnostic explainability technique capable of producing local and global feature importance scores faster than other common state-of-the-art xAI techniques, with no decrease in explanation quality, using counterfactuals.

Chapter 2

Background

2.1 Credit Risk Management

The practice of reviewing loan applications is one with a long history, but the modern practice of risk management is actually fairly recent. In the past, banking was much more localised and loan applications were often based on personal knowledge of the customer and their community reputation. As the world became more connected through global trade, banks gradually grew in size to the point that credit risk was no longer an intuitive concept and older ideas of finance were forced to change in the face of globalization and financial instability such as the Great Depression. Regulatory oversight for banks is a concept less than a century old, and our modern definition of credit risk only emerged in the 1970's and 80's with the birth of the first risk models and risk departments in major banks [32]. The 1988 Basel Accord was a pivotal moment in this development, introducing minimum capital requirements for banks of at least 8% of their risk-weighted assets, improving economic security [22]. Credit risk evolved further towards the turn of the millennium with new regulations such as the 2004 Basel II which further enhanced regulatory oversight and addressed other risk factors such as operational risk [22].

Formally, Credit Risk is defined as the probability of a customer defaulting on their loan, and there are many methods of analysis used to define this probability. Managing this risk is an important task for any financial institution, and as such there is a rich field of research on credit risk dating back to its inception. Traditional approaches to credit risk management rely upon human analysis using the available details about the individual or company who are applying for the loan. Specific rules and policies are used for this purpose, such as the CAMPARI template - Character, Ability, Means, Purpose, Amount, Repayment and Insurance [9].

Character here refers to the responsibility and integrity of the applicant, and is usually judged by reviewing their credit history or credit rating to get an indication of how they have handled credit in the past. Credit rating (or score) is a score given to individuals and companies that reflects their ability to pay back loans and avoid going into default, and it is a key component of modern credit risk. Different countries use different credit scores and so there is no single definition, but a good example of the components of a credit score can be found in the FICO score, commonly used in North America, which combines different factors in a weighted score [48]. 35% is accounted for by payment history - records of past credit repayments, defaults or bankruptcies. Another 30% is covered by amounts owed - a person's current debts. Credit Utilization Ratio is a metric that can be used here, and is in general a helpful score for CRM. It is defined as the ratio of current credit balances to total credit limit, the lower the risk. The remaining 35% consists of the length of credit history, new credit accounts and types of credit used by the customer. Together, this information forms a score. In the case of FICO, the score ranges from 300 to 850, the higher the better.

Capacity refers to the applicant's income sources and stability, and is used as a measure of how likely they are to be able to repay the loan, whereas capital refers to money paid immediately, such as a deposit when renting accommodation. Debt-to-Income-Ratio, the ratio of total debt repayments to gross income, is a very common statistic used when analysing financial security and loan capacity. Collateral is other assets that can be offered in the event of defaulting on a loan, such as property. Loan-to-Value-Ratio is a metric that can be used to measure the appraised value of this collateral, and is defined as the ratio of loan amount to appraised value of property. The lower this score, the better the loan is secured by the collateral, and as such has lower risk. Lastly, conditions refer to the loan itself such as the interest rate, the amount borrowed, and the purpose of the funds. These properties and associated scores define the main types of information banks use in traditional credit risk models, though the specifics differ from bank to bank and location to location. In summary, statistical information about a customer's financial situation is key for making loan decisions in the modern day, which makes it a perfect application area for machine learning.

2.2 Explainable AI

The idea of explainable AI systems is more relevant now than ever before, but the history of the field goes back to long before deep learning. Beginning in the 1970's, 'expert systems' were created as a way of emulating the decision-making process of a human expert through logical inference. These systems were rule-based, relying on a knowledge base containing many different logical rules to be followed when coming to a decision. By then applying inference across these rules, an expert system could come to a conclusion and supply its user with the rules that led to its decision, making it fully understandable and trustworthy since the rules used were designed by human experts [11]. One of the earliest examples is MYCIN, a system developed during the 1970's at Stanford University and used to diagnose bloodstream infections using a knowledge base of around 600 rules [55]. The trend of rule-based AI systems continued to rise throughout the 70's and 80's with more powerful systems, but by the 1990's it became clear that there were limits to this approach. Neural networks were being studied at this time, including research on the possibility of exposing their decision-making process through the extraction of rules, similar to other rule-based systems of the time [59]. However, it was not until the 2010's and the rise of deep learning that the modern field of xAI began to form.

Deep neural networks and similarly complex models are black-boxes with incredible performance but no transparency by default, and ethical questions arise when applying them to sensitive areas such as healthcare, law, or the subject of this paper - finance. By revealing the factors that lead to a model's decision, we can more easily spot biases and other unwanted decision weighting. A normal user of the system can be given understandable explanations, and researchers can utilise these explanations to improve the model. Most importantly, technology can only be relied upon when its human users trust it, and as AI becomes more prevalent and powerful the value of transparency and trust cannot be understated.

2.2.1 Terminology

As the field of explainable AI develops, many new terms are used to refer to common features and requirements. To aid in clarity and understanding, several important concepts are defined below which will be consistent throughout this paper.

2.2.1.1 Transparent vs Explainable Models

A model is considered *transparent* if it is intrinsically understandable by a human with no adjustments. Decision trees and logistic regression are examples of this, as they inherently provide a visual explanation of decision making. These models can also be described as *white-box* models, but it is important to note that the two terms are not interchangeable. Transparency is a property of human understanding of the model's decision making, whereas the definition of a white box is a model whose parameters and architecture is known. Transparent models are sometimes also referred to as *interpretable* models, or *ante-hoc* approaches.

A model is considered *explainable* if it is not intrinsically human-understandable, but utilises an xAI algorithm to provide additional information explaining its decision-making, such as importance of various features to a decision made. Explainability is usually a desired property of *black-box* models, which are models whose parameters and architecture are hidden from the user, such as in Large Language Models like ChatGPT. However, similarly to the above case, the two terms are not interchangeable. Approaches that rely on analysis of the model after training are also known as *post-hoc* approaches.

2.2.1.2 Global vs Local Explanations

A *local* model explanation is one that aims to explain a particular decision made by the model - why it provided a certain output given this input. In contrast, a *global* model explanation is one that analyses the model without any regard to specific predictions, and instead aims to provide understanding with regard to all predictions. Taking the example of feature importance, a local feature importance metric provides an understanding of which input features were most important in deciding the output from the model, whereas a global feature importance metric may average all such local feature breakdowns to provide a global view of which features were most important across all decisions made - hence most important overall.

2.2.1.3 Model-Specific vs Model-Agnostic Explanations

An explainable AI technique is *model-specific* if it can only be applied to a certain kind of model and does not work for all situations, for example an xAI method reliant on the presence of neurons can only be applied to neural networks. The opposite case where a technique is *model-agnostic* implies that it is applicable to all learning problems, which usually means it is reliant on input or output data samples instead of internal model structure.

2.2.1.4 Trustworthiness

Both inherently transparent and explainable models improve a model's *trustworthiness*, which is a measure of human confidence in the model's decision-making ability. Transparent and white-box models are inherently more trustworthy, and one goal of explainability methods is to improve the trustworthiness of black-box models to the same level. This is a metric based on human thoughts, and so trustworthiness is only measurable when human trials have been performed and feedback gathered.

2.2.1.5 Bias and Fairness

Since explanations provide insight into a model's decision process, they are helpful in detecting *bias* - an imbalance in the model's learned weights towards certain subsets of the data. This bias could be inherent to the training data itself and reflect human biases, or come from issues in the model itself such as overfitting. Even if protected traits are removed from a dataset, the model may still form a prejudiced opinion by learning from other features that are more common to a certain subset. Improving *fairness* of models is one of the major goals of explainable AI, allowing for biases to be detected and corrected.

2.2.2 Transparent Models

Some machine learning models are naturally transparent and explainable, such as linear or logistic regression, decision trees and SVMs. Models like these saw much use in the past [26] [65] [8], but with the exception of tree-based learning, their performance is insufficient in the modern day. These models do not require any additional explainability techniques, since a human user can understand the model's decision-making process simply by observing a decision boundary, at least in the case of low-dimensional data, or use techniques like Principal Component Analysis to visualise higher-dimensional data. Decision trees are highly explainable by their nature, as humans are very used to the rule-based tree structure, especially in finance where rule-based approaches are trusted and traditional. These white-box techniques are considered a goal for explainable AI to match. In some cases explainability can be obtained by simplifying a complex model to a transparent one, called a surrogate, covered in the Chapter 3.

2.3 Transformers

In 2017, the field of deep learning was revolutionised by the now-famous paper "Attention is All You Need", which proposed the multi-head attention mechanism and Transformer model [61]. Whilst primarily developed as an advancement in natural language processing aimed at replacing the LSTM-based systems of the time which suffered from small context windows and lack of parallelization, it not only became a new standard for language models but also transformed the fields of computer vision, speech processing and deep learning research as a whole. The Transformer model also led to the later development of BERT and GPT, and is the core technology behind the current advances seen in recent versions of the well-known ChatGPT. It is no exaggeration to say it is the foundation of the current AI boom, and a subject of great academic interest.

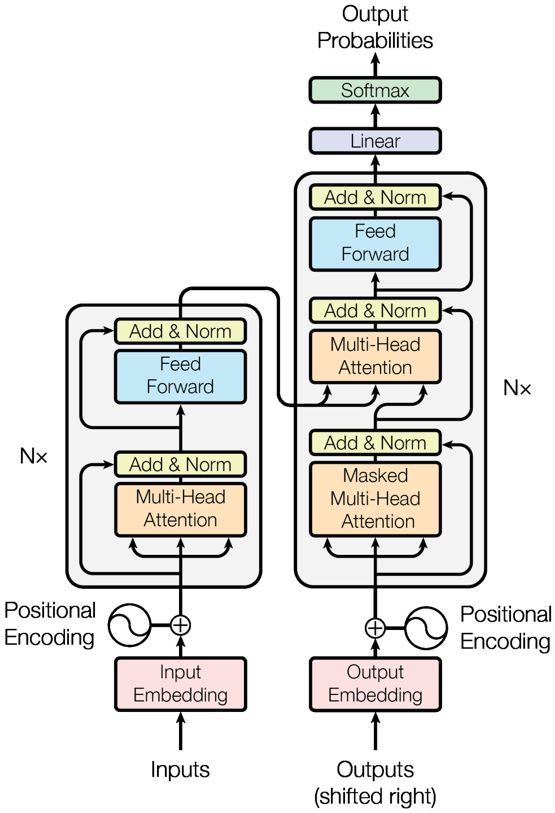


Figure 2.1: The original Transformer architecture proposed by Google researchers [61].

The Transformer is based on the idea of sequence-to-sequence learning, consisting of an encoder and a decoder module as well as *attention* - a way to represent how important different tokens are to other tokens. The encoder processes input data and learns an internal representation or context, and the decoder uses this information when producing output. The architecture can be modified to suit various inputs, hence it's widespread application across different fields. The attention mechanism is the key component of both the encoder and decoder - it takes in the original input vector (in the language modelling case, these are positionally-encoded text embeddings) as query, key and value. By multiplying the query and key vectors and applying a softmax function, we can obtain probabilities for each query-key pair, such as two tokens in a sentence. This is referred to as how strongly one input token 'attends to' another - in other words, the relevance of all input features to all others. By finally multiplying by the original values, the original embeddings are modified to encode this relationship information which can then be learned from. Formally, attention can be written as:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is the key dimension, equal to the dimension of the model's learned embeddings d_{model} divided by the number of attention heads h. This type of attention is also called self-attention. In the original paper, they use 8 attention heads with $d_k = 64$. Dividing by $\frac{1}{\sqrt{d_k}}$ is used to stabilise the model and counteract vanishing/exploding gradients in the softmax function. A single instance of attention is referred to as one 'attention head', and the Transformer applies attention many times over different linear projections of size d_k, in order to learn different relationships, hence 'multi-headed'. Lastly, in both the encoder and decoder, a feed-forward neural network layer is applied to further transform the rich, contextual information obtained from the attention mechanism and learn complex relationships between input features. Transformer blocks are often stacked sequentially, with each additional block providing more intricate knowledge at the cost of huge additional numbers of parameters.

Chapter 3

Literature Review

3.1 Machine Learning for Credit Risk

3.1.1 Early Approaches

The pre-ML process of credit risk management essentially involved recording important features about the applicant and testing them against a set of known rules to produce an outcome. The resources used to make a credit decision and the outcome of that decision could be gathered into a dataset and analysed to improve the bank's rules such as raising or lowering certain decision thresholds. Therefore, banks possess massive quantities of such loan data, which became particularly valuable for machine learning models, allowing for training of larger, complex models capable of taking much more relevant information into consideration than a human being.

The earliest kinds of models, studied over the late 90's and 2000's, included basic neural networks, linear/logistic regression, kNN and decision trees, as well as SVM models towards the mid to late 2000's [36] [26] [30] [5] [65] [35] [19]. At this time, the most promising models were Logistic Regression [8], SVM [35] and hybrid models [40] [60] that combine multiple classifiers into one - a trend that continues to be relevant today. These models are also naturally transparent and interpretable - LR and SVM models create understandable visual decision boundaries, and decision trees can be thought of as an algorithmically generated set of rules to follow one after the other, similar to traditional banking techniques, meaning there were few issues implementing them in practice and can be thought of as good traditional baselines.

3.1.2 Bagging and Boosting

By the late 2000's and early to mid 2010's, the above models were outperformed by more complex approaches like random forest [49] which improved upon the accuracy of models such as tuned logistic regression and could be trained relatively quickly [34] [45]. Decision trees had been researched for a long time, but were usually outperformed by other approaches when used alone due to their high variance. However the technique of 'bagging', or training multiple trees and averaging their results, improved performance to state-of-the-art levels. Towards the late 2010's, extreme gradient boosting [17] showed even greater potential [14]. In contrast to bagging, boosting is the process of sequentially training small trees that are weak, biased learners on their own. However, by having each subsequent tree aim to reduce the residual error in the current ensemble, over time the overall ensemble becomes a strong learner. XGBoost is an optimised, enhanced gradient boosting model that improves upon several of the weaknesses of traditional gradient boosting. It has superior efficiency due to utilising Hessian information, makes use of parallelisation techniques to speed up training times and offers L1 and L2 regularization to better prevent overfitting, making it an attractive tree-based model for machine learning. On tabular data like credit risk, it has outperformed all other approaches and is the target to beat for competing approaches [51] [63].

3.1.3 Neural Networks

Neural networks have also seen use in credit risk literature for over 20 years, but older research focused on multi-layer perceptron models with only one or two hidden layers [66] [5]. Modern deep learning approaches achieve greater performance by stacking many hidden layers. Earlier layers capture low-level features and with greater depth comes more non-linear combinations, allowing for superior generalization to complex functions and greater expressive power [23]. Several papers achieve results surpassing the traditional approaches such as LR, SVM and shallow neural networks [21] [44]. Sampling techniques such as oversampling (SMOTE) and random undersampling are widely used to improve accuracy in the field of credit risk, since such datasets often contain class imbalance where the majority of customers do not default [39]. However, tabular data remains one of the few types of learning problem in which neural networks have not become dominant and are outperformed by tree-based models. Grinsztajn et al. [29] highlight some reasons for this difference, such as how neural networks are biased to overly smooth solutions, and how they are not robust to uninformative features which are common in tabular learning problems.

3.1.4 TabNet

To narrow the gap between neural networks and tree-based models, a unique approach is needed. TabNet [4] is a novel deep neural network architecture combining the strengths of trees, neural networks, and Transformer concepts such as attention and its encoder-decoder-style design. Data is processed across several ‘decision steps’, and a sequential attention mechanism is applied at each step. This allows the model to select a subset of features that are semantically meaningful, similar to how a decision tree splits features at each node. Additionally, this way of processing data provides native explainability, as the model captures which features were important to the attention mechanism at each decision step and uses a simple aggregate function across steps to define feature importance masks. By analysing feature importances, Arik et al. show how TabNet is able to focus on relevant variables while disregarding irrelevant ones using both synthetic and real-world datasets, achieving more appropriate importance scores compared to other methods such as LIME whilst retaining high test accuracy, making it an attractive solution for explainable tabular data classification. Liu et al. [41] applied TabNet to the problem of credit risk, showing good performance compared to gradient boosting techniques on their dataset. However, their evaluation is relatively weak and difficult to compare to other papers as they do not use the common AUC score in favour of their own metric. It provides a promising proof of concept, but would require additional research and more thorough evaluation.

3.2 Explainable AI

3.2.1 Surrogate Models

LIME [53] is a local, model-agnostic algorithm introduced in 2016 that treats the model as some black box function f, for example a perceptron function $f(x) = Wx + b$. It generates samples in the local neighbourhood of a point of interest near the decision boundary, evaluates them using function f, then fits an interpretable model to these local points by using the complex model as a source of truth and training on this local dataset. The resulting model, for example a logistic regression classifier, produces interpretable results as it is naturally transparent, but these results are informed by the black box model's knowledge due to being trained on its classifications. Its model-agnostic nature makes it widely applicable and LIME has seen use across many domains such as healthcare and finance, and is seen frequently in credit risk literature [13] [46].

Neural Additive Models or NAMs are another surrogate technique proposed in 2021 [2]. NAMs combine the features of deep neural networks and Generalized Additive Models [31], a type of regression model that replaces the linear predictor with a sum of functions of the form $g(\mathbb{E}[y]) = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$ where each f_i is a smooth function of the predictor variable $x_i \in X = (x_1, x_2, \ldots, x_n)$. In NAMs, each f_i is a neural network that attends to just that variable, and by training many subnetworks jointly and summing their outputs like in GAMs the resulting model has a high predictive accuracy and is interpretable unlike typical DNNs. Each surrogate sub-network can be visualised as a graph to detect bias, allowing for both global interpretations and local explanations, however this technique is model-specific to neural networks.

3.2.2 Feature Relevance

Many xAI techniques present explainability by obtaining importance scores. In other words, they ask the following question: "How strongly does the value of each input feature influence the model to make a certain decision?" One such technique is counterfactuals, a concept originating from other fields like philosophy and economics, but one that can applied to ML models as a local explanation technique [62]. A counterfactual explanation is one that explains the minimal change required for a different outcome, and they come in the form "Decision A was made because variable X is less than Y. If X was greater than Y, decision B would have been made." In xAI, a counterfactual explanation is similarly defined as the smallest change in the input features that changes the prediction outcome, hence providing the user with the reasoning behind a model's decision. This can be done by optimising a modified objective function that uses the distance between an original prediction x and the counterfactual x'. However, one issue with counterfactuals is that they suffer from the Rashomon Effect - by presenting a human user with several counterfactuals which are all equally valid but possibly contradict each other they will be confused. Additionally, this same method can also be used in the generation of adversarial attacks [24], since by making a tiny modification to input data the entire classification is changed - the modified input may look almost identical to a human, but functions almost like an optical illusion to a machine learning model. The same technique used to create counterfactuals can be used to fool them, and to combat this vulnerability and improve trustworthiness, more robust counterfactuals must be generated, especially in critical decision-making applications like credit risk [56].

Partial Dependence Plots or PDP are a global, model-agnostic explanation technique that make use of feature relevance to explain predictions [25]. A PDP is generated by choosing a set of values of interest for one feature, for example [0, 1, 2, 3, 4], then setting the feature of interest to these values and calculating the average output for all data points, leaving all other features alone. This information can then be plotted as a graph of how the model's predictions are partially dependent on this feature, and by repeating for all features a global explanation is obtained. PDPs are useful for any machine learning task due to their global, model-agnostic nature, and such plots appear in many different application domains, including industry [27] and most relevant, credit risk models [7] [57].

Lastly, but of significant importance, is SHAP [42], a popular local, model-agnostic explainable AI technique that can be used to calculate feature importance for a given prediction. The technique is based upon the game theory concept of Shapley Values, a technique to calculate the individual contributions of a player(s) to a team using their average marginal contribution across all possible subsets of players. In the realm of machine learning, SHAP performs the same marginal contribution calculation where the 'players' are features, essentially calculating how strongly the presence or absence of certain features contributes to the model's decision. This is expensive to calculate as it involves iterating over all possible feature subsets, but techniques such as Kernel SHAP exist to mitigate this problem [42]. SHAP is often used in tandem with or instead of LIME, as both are local, model-agnostic techniques, though SHAP is a more consistent, stable approach with favourable theoretical properties and hence sees more use in recent research. In credit risk literature, it is an extremely popular method for adding explainability to any model - possibly the most widespread approach and a major target to beat [46] [12] [27].

3.2.3 Convolutional Techniques

While the use of Convolutional Neural Networks (CNNs) for credit risk may seem unintuitive due to the data not being in the form of a 2D image, it is possible to convert tabular credit risk data into this form using feature transformations such as Weight Of Evidence to encode the information [64]. This allows for later usage of image-based explainability techniques whilst achieving high predictive accuracy [20], and CNN-based credit risk models, including hybrids that combine a CNN with other models, are an ongoing area of research [68]. Many such techniques exist to highlight the relevant pixels to an output in image-based machine learning, of which I will cover two of interest.

One such technique is Layerwise Relevance BackPropagation (LRP) [6], a local explainability technique used in CNNs that provides relevance scores for a certain prediction by identifying which pixels caused the strongest activations in the network. It achieves this by starting from the output neuron containing the relevant class score then backpropagating this score to the previous layer. The strength of connections between each layer of the network and its previous layer is calculated iteratively, until it reaches the final convolutional layer, hence displaying the relevance of each pixel to the final decision. Since it relies upon the CNN structure it is a model-specific technique, but backpropagation is easily applicable to both the dense MLP and convolutional layers of any CNN and is an intuitive process in machine learning, and also one that is not limited to just convolutional networks as any neural network can take advantage of backpropagation.

Another widely-used algorithm for image-based explainability is Gradient-weighted Class Activation Mapping, GradCAM [54] which performs a similar gradient-based approach. Unlike LRP however, GradCAM backpropagates the gradients from the final output prediction for a class of interest, denoted as y^c, to the feature maps of the last convolutional layer, denoted as A^k, instead of all the way back to the input layer. Here, it generates importance weights by calculating the gradients $\frac{\partial y^c}{\partial A^k}$ for each feature map k, then averaging them:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

where Z is the number of pixels in the feature map, $i \times j$. Once the average gradients for each feature map are calculated, GradCAM is then defined as the weighted sum of feature maps passed through a ReLU activation function to only retain the positive values that we are interested in:

$$GradCAM = ReLU(\sum_k \alpha_k^c A^k)$$

Compared to LRP, GradCAM is less computationally expensive, and creates a higher-level heatmap of regions of interest instead of fine-grained pixel-level details. It is most often used in medical research, but can be applied to any 2D data and produces good, human-interpretable representations [33] [67].

3.3 Credit Risk Transformers

Whilst the Transformer was originally only intended as a language model, it's architecture is widely applicable to many deep learning applications when modified. Overall, compared to the volume of literature covering models such as XGBoost, there is a small quantity of research on Transformers for credit risk and a clear gap for further research and development. A feature common to all Transformer-based credit risk models is that the Transformer architecture is modified or combined with other systems to improve performance and handle different input features. Zhang et al. [69] propose a model consisting of a Transformer encoder followed by a CatBoost random forest ensemble, and finally a decoder. CatBoost is very similar to XGBoost, implementing the same boosting techniques and optimisations whilst also natively supporting categorical features [50]. Categorical feature support is key to the credit risk domain, as many relevant features are categorical rather than numerical. By combining the unsupervised Transformer's ability to uncover the relationships between input features and the supervised CatBoost tree to improve classification ability, the resulting model aims to improve predictive accuracy and understanding, whilst also being able to handle high-dimensional features at a large scale. They show promising results that outperforms other standard models, as well as the baseline Transformer model, inspiring future work in the area.

Wang et al. provide another detailed study into the use of a modified Transformer hybrid model with their CNN-SFTransformer [64]. Their approach is similar to the hybrid approach implemented by Zhang et al., utilising a modified Transformer block for unsupervised learning and a two-layer one-dimensional CNN for supervised learning, done in parallel rather than sequentially. Feature data is provided to both networks, and their output features fused to produce the final output. Their modified Transformer architecture, SFTransformer, uses a different attention mechanism called semantic feature-based multi-head attention, which takes in its keys, queries and values as raw feature data rather than the usual positionally-encoded input embeddings. To improve feature extraction from data and allow the model to better learn semantic relationships, the concept of Gaussian-weighted distribution tokenization is used, where the query and key are multiplied by different Gaussian weight matrices in each attention head such that each head focuses on different semantic information. The scaled dot product is used here to improve computation, and attention is calculated in the usual way, just using the Gaussian-weighted key and query instead. They provide thorough experimental details and analysis on two popular credit risk datasets that will be discussed in the next chapter, showing improved predictive accuracy over baseline models. Whilst their model does achieve an average 0.1 AUC gain, it is much more complex than the tree-based alternatives and offers no explainability, reducing its practicality.

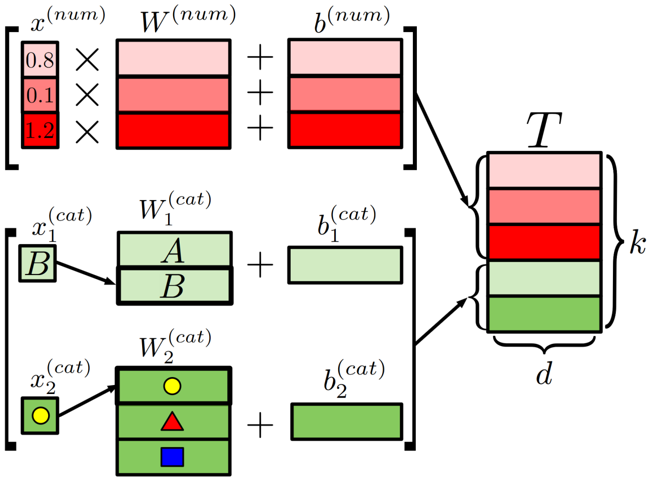


Figure 3.1: The Feature Tokenizer component of the FT-Transformer. Diagram obtained directly from Gorishniy et al. [28]

Another approach to credit risk transformers is the FT-Transformer proposed by Gorishniy et al. [28] in 2023. Unlike other similar models, they retain the traditional Transformer module with no major changes and instead focus on a novel feature encoding method seen in Figure 3.1 above. Their Feature Tokenizer is capable of encoding both numerical and categorical data into embeddings, making it highly applicable to credit risk and other datasets with multiple feature types. Numerical features are encoded using element-wise multiplication with a weight parameter $W^{(num)}$ and bias $b^{(num)}$, whereas categorical features instead use a lookup table $W^{(cat)}$ and bias $b^{(cat)}$, transforming the input with a one-hot vector encoding. They experiment on a large number of datasets and obtain consistent state-of-the-art results, which is even more impressive considering the remainder of the architecture is untouched, however gradient boosted trees remain dominant on some datasets. Overall, their Feature Tokenizer is a highly adaptable module for considerably improving the performance of Transformer-based models, and combining it with other modifications could lead to further improvements.

Lastly, Zhang et al. [68] propose a use for the Transformer encoder that reflects its original purpose as a language model. They combine typical loan information with the textual description of the loan application, and apply the encoder to these descriptions to obtain extracted features that help with classification. Their analysis shows the effectiveness of these new learned features, but similarly to Wang et al. the black-box nature of their model is something they note as a major drawback that would not allow its use in practical financial applications. This issue is one that plagues many Transformer-based approaches to credit risk, and it shows the value of explainability research for complex models like Transformers. In summary, many different approaches have been considered using Transformers for credit risk management, but no approach has found dominance and there is a broad lack of interpretable models in the area.

3.4 Explainable Transformers

The most natural approach to understanding Transformers is to understand self-attention, the core of the model, and a large portion of literature falls into this category. A foundational work in this area is the concept of attention rollout, or attention flow [1], which operates similarly to LRP by using attention weights as a score of relevance and backpropagating from the final embeddings to the input tokens. By assuming a linear combination of input tokens throughout the layers of a Transformer encoder, they show that this method is much more interpretable that raw attention. However, they draw attention to the simplifying assumption of their interpretation of these attention weights as a source of caution, and Chefer et al. [16] later identified that attention rollout fails to distinguish between positive and negative contributions, and attention flow is too slow for large-scale use. They instead employ a full LRP-inspired approach, using gradient integration for self-attention layers and achieving superior results for Transformer encoders. Their initial work is not applicable to all Transformers, but is improved upon to achieve state-of-the-art results on self-attention, co-attention and encoder-decoder attention [15]. This LRP-based approach is one that has seen a large amount of research interest, and continues to see improvement such as Ali et al. [3] introducing a more stable and reliable LRP approach for Transformers. They identify two issues with previous gradient-based approaches where the property of conservation is broken and input variables are incorrectly scored. They show that this property breaks in two areas, the attention heads and layer normalisation, and define new conservation rules for these components that achieve better conservation across layers and state-of-the-art AUAC (Area Under Activation Curve) scores. These approaches are applicable to any Transformer but were designed for natural language or vision purposes, but attention has also seen use in tabular and categorical data like the kinds seen in credit risk datasets.

Aside from utilising attention, several other methods have been proposed to incorporate explainability into Transformers. Like with neural networks, surrogate models and post-hoc local explanations exist for Transformers - LIME and SHAP are both applicable due to their model-agnostic nature and used in many different explainable Transformers, but Leeman et al. [37] provide proofs that Transformers cannot represent additive models and cast doubt on the applicability of these popular techniques. They instead propose the Softmax-Linked Additive Log-Odds Model (SLALOM), a surrogate model explanation technique capable of representing Transformer output more closely than a GAM where feature explanations can be obtained in a 2D space. While not directly applicable to tabular data, they provide an interesting argument against the use of other popular surrogate models. Thielmann et al. [58] propose a hybrid approach for tabular data inspired by NAMs, using a Transformer only for categorical features and independent neural networks for each continuous feature. Like in NAMs, the overall prediction is obtained by concatenating outputs. Each continuous feature can easily be visualised as a graph, and categorical features can be interpreted through the use of attention-based methods such as those described above, with only a minor performance tradeoff. Another alternative approach to explainability in transformers is the use of perturbation - applying noise to the input and observing changes in output, similarly in theory to a PDP. Rao et al. [52] apply this technique to heart failure prediction using BEHRT [38], developing a local surrogate approach that uses perturbations to measure the importance of individual features on a certain prediction. They also show that by aggregating these results across the full dataset, a global explanation can be obtained and overall relevance contributions analysed. Whilst their approach focused on time series health data, this perturbation-based approach to feature importance could easily be explored within credit risk Transformers. In summary, I have identified that the key research areas in Transformer explainability are attention-based approaches that focus on the Transformer internals, GAM-inspired hybrid approaches, and surrogate or local methods that follow the model-agnostic approach and focus on the features themselves.

Chapter 4

Preliminaries

The machine learning task I will explore is credit risk (loan default) prediction on tabular data. Each data point will consist of some amount of variables, more commonly called features, with many such data points making up the full dataset. The features in the data will be details about a customer that are relevant to deciding if they should be given a loan or not, such as the kinds highlighted in Section 2.1. These could be personal details like age or income, details specific to the loan they are requesting such as the amount or collateral, or records of current/past loans and how they were repaid. Notably, credit risk datasets are often mixed with both numeric (real-valued) and categorical (integer or string) data, which can pose a problem for traditional classifers which cannot handle categorical input, instead working exclusively in the space of real numbers. The type of classification we wish to perform is binary classification - predicting either 0 (safe loan) or 1 (risk of default) for each data point. A good credit risk model should be able to train on many examples of this data, then accurately predict this binary label for previously unseen data.

More formally, we have a dataset (X,y) where X is a matrix of n samples, each consisting of k features (or other words, a $n \times k$ matrix) and y is a corresponding list of binary labels. Additionally, each $x_i \in X$ is a vector that contains both numerical and categorical data, therefore input data must either be encoded into a suitable form prior to processing, or the model itself must have some internal representation of categorical features that it can use to its advantage. A common method of handling such categories is a simple ordinal encoding, translating each category to a whole number such as Category A = 1, Category B = 2, etc. First, as with any machine learning problem, the full dataset must be split into training, validation and test sets. The training set is used to train the model, the validation set is used to adjust hyperparameters, and the test set is only used to record final performance. During training, a loss function is used to compare the model's current predictions to the true labels, then adjust its internal representation to reflect the loss, with a common choice being Binary Cross Entropy Loss. In order to correctly classify unknown future samples in the test set, the model must learn the complex relationships between input features and identify values that represent higher probability of default. The performance of this model can then be tested against any common evaluation metric, detailed in the Experiment chapter.

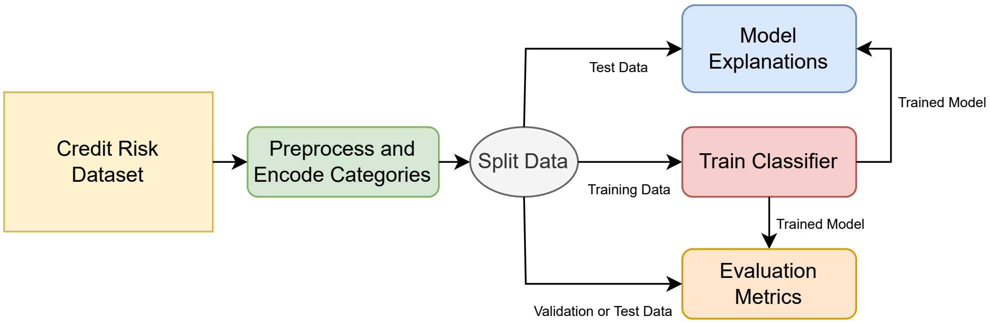


Figure 4.1: The typical process of training and evaluating a credit risk model.

The final step in the process is explanation, a step not always present in traditional ML but an important one for credit risk models, for reasons detailed in the previous chapters. Here, the trained model and test set are provided to whatever explainability algorithm(s) are to be used, which are typically model-agnostic methods so that any kind of classifier can be provided. The end result of the entire process is performance metrics evaluated on either validation or test data, and feature importance scores, usually in the form of a bar chart or other visualisation.

Chapter 5

Method

5.1 XGFT-Transformer

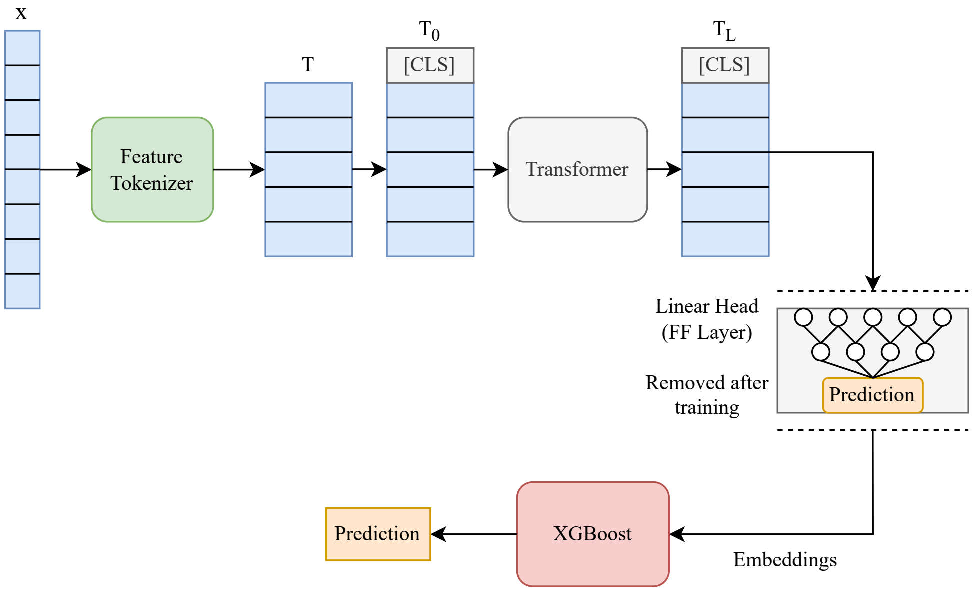


Figure 5.1: Architecture of the XGFT-Transformer.

The XGFT-Transformer is my novel hybrid model aimed at combining the representative power of stacked Transformer units and the historically high classification performance of XGBoost, inspired by similar research into hybrid Transformer models such as by Zhang et al [69]. Existing research on such models uses the original Transformer input encodings, but the Feature Tokenizer (FT) module introduced by Gorishniy et al. [28] has shown promising performance improvements for tabular data compared to the traditional Transformer encoding technique.

The model consists of two separate components trained with separate loss functions. First, the FT-Transformer is trained as normal, resulting in an encoder stack with a powerful internal representation of the dataset, already capable of high performance on its own. Next, the encoder is *beheaded*, removing the final fully-connected layers (Linear Head) of the FT-Transformer, transforming it from a model that outputs binary probabilities to one that produces raw embeddings. This allows us to subsequently encode the training data using the beheaded model, obtaining a transformed training set of embeddedings that represent the Transformer encoder's classification knowledge. This new training set is finally used as the input to the XGBoost model, which is trained as normal to obtain the binary probabilities for classification, with the training loss only propagating through the XGBoost model and not the Transformer itself. During inference, validation or test data is fed forward through the beheaded encoder stack and into the XGBoost model as one fluid process. By providing XGBoost with a dense encoded representation of useful semantic knowledge instead of raw features, it may be able to learn more efficiently. Additionally, each half of the model can cover the other's weaknesses, with the goal of increased stability and less variance in performance. Unlike other hybrid Transformer models, a decoder module is not used, which reduces the number of parameters and leads to faster performance. This approach is also highly adaptable, with any other classifier able to replace the role of XGBoost, such as similar boosted tree models like CatBoost.

5.2 Global Counterfactual Importance

My novel approach to model-agnostic explainability is inspired by counterfactuals, a less common xAI technique. I identified multiple issues with raw counterfactuals in Section 3, however my novel algorithm applies a fix for all of these issues by using counterfactuals to inform a global importance score. By returning one global feature importance metric we can avoid the Rashomon Effect. Adding a small amount of random noise as suggested by Slack et al. [56] can reduce the counterfactuals' vulnerability to adversarial classifiers, but this is left to be explored in future work. Additionally, my algorithm is capable of handling both numerical and categorical variables with greater user control over the algorithm, making it more applicable to tabular problems like credit risk. The pseudocode for Global Counterfactual Importance (GCI) can be seen below:

Algorithm 1 Global Counterfactual Importance

```
Require: Model M trained on dataset $(x^{train}, y^{train})$, test set $(x^{test}, y^{test})$, Categorical
  Scaler $k$, Largest Distance $d_{max}$
  $D \leftarrow \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$
  $C \leftarrow$ [best counterfactual(M, x) for x in $x^{test}$]    ▷ Obtain C from existing algorithms
  for $c \in C$ do                                    ▷ Calculate a score for each counterfactual
    $d^c \leftarrow$ normalizedDistance(c, $x_i^{test}$)
    $d^c \leftarrow [f \times ||d^c||$ for $f \in d^c]$               ▷ Weight by size of $d^c$
    $D_i \leftarrow d^c$
  end for
  $D \leftarrow$ D.applymap($x \leftarrow |x - d_{max}|$ if $x > 0$ else x)          ▷ Rescale values
  return mean(D)
```

Algorithm 2 Normalized Distance

```
Require: Counterfactual $c$, Test instance $x^{test}$, Categorical Scaler $k$
  $d^c \leftarrow [0, \dots, 0]$
  for $f \in c$ do
    $x_{min} \leftarrow$ min(range(f))                    ▷ Min/Max value of feature f
    $x_{max} \leftarrow$ max(range(f))
    $x_{original} \leftarrow \frac{x_f^{test} - x_{min}}{x_{max} - x_{min}}$                     ▷ Normalize to range 0-1
    $x_{counter} \leftarrow \frac{c_f - x_{min}}{x_{max} - x_{min}}$
    $d_f^c \leftarrow |x_{original} - x_{counter}|$
    if $f$ is categorical then
      $d_f^c \leftarrow d_f^c \times k$                   ▷ Categorical importance scaling
    end if
  end for
  return $d^c$
```

One of the main difficulties in creating a unified score for data which can contain both numerical and categorical features is how to represent a measure of distance between vectors. To quantify how much of a change was required to flip the model's classification, we want to be able to measure the distance between its original position and the position of the counterfactual in vector space. However, whilst the numeric features operate in a real number space, the categorical features are restricted to n whole numbers. To alleviate this issue, each feature f is normalized, scaling each to range between 0 and 1, with 1 being the maximum value of that feature, which then allows for a simple Euclidean distance measure to be taken between the two points. An important feature is considered to be one that a) only needs to be changed slightly for the entire prediction to change, and b) makes up a high percentage of the counterfactual change. To achieve this logic, each feature f in the distance vector is scaled by the absolute value of the full distance vector. If both f and $||d^c||$ are small (which also means f makes up the majority of the counterfactual), the value becomes even smaller, whereas the opposite is true for large values. Unlike other feature importance scores, a smaller value implies greater importance, but to conform with convention the scores are scaled such that small values become large and vice-versa, with scores of 0 remaining at 0. To ensure scores are comparable between runs, hyperparameter d_{max} should be set to a high enough value such that no counterfactual distance will be greater than it - a static value of 10 is used in my experiments. The resulting matrix contains local importance scores for the full test set, but these are finally aggregated with a simple mean to obtain Global Counterfactual Importance scores.

With the default settings of the algorithm, categorical variables are treated more harshly due to normalization. For example, a binary category being changed from 0 to 1 would have a distance of 1, equivalent to a numerical category increasing from its minimum to its maximum value. This means a categorical feature changing is seen as less important overall, since any change to it is perceived as a large change. Rather than a flaw however, this behaviour is one that can be controlled and adjusted to suit the application. For settings where a categorical shift represents an important feature, the hyperparameter k (1 by default) can be adjusted. Small values of k shift the balance to favour categorical variables, shrinking their distances, whereas values greater than 1 do the opposite. Therefore, GCI allows for greater user control over what should be perceived as a more important change than other xAI algorithms for mixed tabular datasets. It can be applied to any application domain and adjusted to suit the data, making for a lightweight, flexible algorithm capable of producing feature importances in the vein of other model-agnostic methods like SHAP.

Chapter 6

Experiments

6.1 Datasets

The project uses two credit risk datasets, detailed below, both obtained from the UCI Machine Learning Repository. Both are tabular datasets containing numerical and categorical features for each data point, a setup that is traditionally difficult for some classifiers like neural networks. In the following sections I will describe their structures and detail any preprocessing steps applied to the data before use.

6.1.1 German Credit Risk Dataset

The first dataset is a German credit risk dataset, chosen for its historical prevalence in literature. This dataset has been used for decades as a standard benchmark, making it an excellent source for performance comparisons between the classifiers implemented in this paper and other prominent works in literature. It contains 1000 data points with 20 features each, 8 of them numerical such as loan duration and age, and 12 categorical such as loan purpose and checking account status. The data is labelled with the positive class 2 being those at risk of default, and then negative class 1 being safe loans. The dataset is available both with raw categorical data and numerical substitutions, but I will be using the raw categories and performing encodings manually to account for both models which require categorical indexes and those that cannot handle them. This dataset was lightly preprocessed, changing the class labels to 0 (safe loan) and 1 (defaulter) to reflect the standards of binary classification. A class imbalance was indentified within the dataset, with 700 instances considered to be good loans, whereas 300 are considered to be at risk of defaulting. To analyse the effect of class imbalance, two versions of the dataset are used - the base version, and an oversampled version. To perform this oversampling, I use Synthetic Minority Oversampling Technique or (SMOTE), specifically the SMOTE-NC variant which allows for categorical variables in the data, resulting in an additional 400 synthetic positive data points and raising the total dataset size to 1400. This is still considered a small dataset by modern ML standards, and the larger, complex classifiers may struggle due to its size, a factor I will explore in later analysis.

6.1.2 Taiwan Credit Risk Dataset

The second dataset is newer, obtained from a Taiwanese bank, chosen for its larger size. Whilst this dataset has not seen as much usage in literature compared to the first dataset, it is much more appropriate for modern, large models due to containing 30x the amount of data points. It contains 30000 instances with 23 features, 9 categorical such as education and payment history and 14 numerical such as bill statement amounts and corresponding payment amounts, and a binary label where 1 corresponds to a defaulter and 0 is a safe loan. This data came preprocessed with categorical data represented as numbers with associated labels, but as some classifiers used in this project require categorical data it was first re-categorised with appropriate strings representing the numerical substitutions. Then, like with the first dataset, case-by-case encodings are applied depending on the classifier's needs. Unlike the German dataset which focuses on personal qualities, the Taiwan dataset includes a history of six months of loan details, meaning the classifiers will have to learn very different features to adequately represent both datasets. This dataset contains 6636 positive (defaulter) samples and 23364 negative (good) samples, showing a similar class imbalance as the German dataset as most people do not default on their loans. This could still pose a challenge for classifiers, but with 20 times more defaulter data than the German dataset the effects should be lessened.

6.1.3 Final Data Preparations

After importing the datasets, they are split into training, validation and tests sets. For evaluation using an validation set, the data is split at a ratio of 75-12.5-12.5, whereas for k-fold cross-validation a common split of 80-20 is used. Due to the small size of the German dataset, cross validation is a valuable technique that allows for the use of more data during training.

6.2 Credit Risk Models

The following models were implemented as a baseline for performance and explainability comparisons:

- **Logistic Regression** - The simplest model, used mostly as a sanity check. Logistic regression is also a transparent model, but explainability techniques can still be applied to it such as counterfactuals. The Scikit-Learn python library was used for the implementation of this model.
- **XGBoost** - A popular model in credit risk literature, XGBoost is an optimised implementation of boosted trees and is an excellent baseline for comparison of more complex models as it and similar tree-based models are a current standard for tabular data classification. The experiments use the official XGBoost python library.
- **TabNet** - An attentive neural network-based model for tabular data. TabNet has been used for credit risk classification in literature before, but results are sparse. Neural networks traditionally perform poorly on tabular data, but this model was included in order to evaluate and compare its performance more deeply than in other works. This model makes use of the categorical data internally, and no initial encoding is used. The experiments use the PyTorch-Tabnet library for implementation of the model.
- **FT-Transformer** - This Transformer model forms a core component of the novel classifier introduced in the next section, and represents the state-of-the-art for tabular data Transformers. I hope to evaluate its performance to gauge the usefulness of larger models compared to the simple baselines such as XGBoost and identify if Transformers are worthwhile to use on tabular problems such as credit risk. Like TabNet, this model makes use of the categorical data internally, not requiring an initial encoding. The experiments use the PyTorch-Tabular library's implementation of the model.

Additionally, for the implementation of the XGFT-Transformer I use both of the corresponding libraries above - XGBoost and PyTorch-Tabular. The latter required me to create a custom tabular Transformer implementation, inheriting their existing architecture and adding the required beheading method and custom predict function.

6.3 Explainability Algorithms

Many kinds of classifier are used in my experiments, and industrial credit risk models may be modified and customised in many unknown proprietary ways. For these reasons as well as its dominance in credit risk literature, I chose to focus on model-agnostic explainability and explore xAI techniques that can be applied to any architecture, making use of model inputs and outputs to obtain feature importance scores at a local or global level. My literature review identified SHAP as the most prominent model-agnostic xAI technique for credit risk, as it has seen widespread adoption in many areas of research and industry. The SHAP Python library is used in my experiments, which implements several versions of the algorithm allowing it to be adapted to any model, such as Kernel SHAP and Tree SHAP. The appropriate version of SHAP for each classifier is used, and Shapley Values are obtained which can be plotted to obtain both local and global importance scores. LIME was also considered for this purpose, but by default it only produces local explanations and is less popular in modern works than SHAP. SP-LIME would allow for global comparisons, but this is left to future work. For the implementation of GCI, the official DiCE [47] library is used to generate the counterfactuals. However, the library was originally only compatible with sklearn logistic regression, so I implemented custom DiCE model representations to fix this.

6.4 Experimental Setup

6.4.1 Hyperparameter Optimization

For each combination of model and dataset as described in the above sections, I use an identical experimental setup. Firstly, all five models undergo hyperparameter optimization using Random Search. This algorithm was chosen as the ideal tradeoff between performance and time required for optimization, though other techniques such as Grid Search and Bayesian Optimization were also tested. A set of potential values for each hyperparameter is given to the algorithm, then the model is evaluated over many iteration on random selections from the set. The exact search spaces used in random search can be seen in Appendix A. In my experiments, 500 iterations of random search are performed to ensure high confidence in the final values whilst retaining reasonable runtimes. To provide the models with as much training data as possible, 4-fold cross validation is used at each evaluation to decide the best hyperparameter selection. Once the optimization is complete, these values are saved to a file to be used in later experiments. The final test scores are obtained by training each model with its optimal hyperparameters, then using the test set predictions to calculate the metrics described in the following section.

6.4.2 Evaluation Metrics

In the case of binary classification of credit risk, the results from a classifier can be in one of four categories. True Positive (TP) for customers who are correctly identified as being at risk of default, False Positive (FP) for customers who are incorrectly identified as such, False Negative (FN) for defaulting customers who are classified as safe and True Negative (TN) for safe customers who are classified as such. The first and simplest evaluation metric used in my experiments is a confusion matrix, which records these values in a 2x2 matrix. Of these values, FNs are the most dangerous and important to be avoided as much as possible. Next, True and False Positive Rate is calculated as below:

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

Using these, the Receiver Operating Characteristic (ROC) curve can be obtained by plotting a graph of FPR on the horizontal axis and TPR on the vertical axis, depicting the trade-offs between the two rates at different thresholds. A random classifier will draw a straight line through the middle (0.5, 0.5) of this space - the higher the curve above this, the better the classifier, and any line below this is worse than random. Area Under the ROC Curve (AUC) is the most common performance metric seen in credit risk literature and also sees widespread use in machine learning overall. It provides a measure of performance across all thresholds and can be interpreted as the probability that a random positive sample will be ranked higher than a random negative one. The Kolmogorov–Smirnov (KS) metric is another common metric seen in credit risk prediction. It is calculated by sorting the samples by predicted default probability, then obtaining the cumulative true/false positive rates and calculating the maximum difference between the two curves, i.e. $max(TPR-FPR)$. The larger this value, the better the model's ability to distinguish between the binary classes. Confusion matrices, AUC and KS metrics are all used as evaluation metrics in my experiments, and where available I also provide standard deviation of results as a confidence measure.

6.5 Results and Analysis

6.5.1 Performance Comparisons

Method	German		German (SMOTE)		Taiwan	
	AUC	KS	AUC	KS	AUC	KS
Logistic Regression	0.734 ± 5e-2	0.419 ± 8e-2	0.814 ± 3e-2	0.528 ± 4e-2	0.707 ± 6e-3	0.326 ± 8e-3
XGBoost	**0.801** ± **3e-2**	**0.495** ± **6e-2**	**0.888** ± **2e-2**	**0.637** ± **4e-2**	0.779 ± 7e-3	0.428 ± 1e-2
TabNet	0.561 ± 3e-2	0.127 ± 2e-2	0.647 ± 3e-2	0.271 ± 3e-2	0.77 ± 8e-3	0.416 ± 2e-2
FT-Transfomer	0.76 ± 3e-2	0.436 ± 6e-2	0.86 ± 3e-2	0.588 ± 5e-2	0.782 ± 6e-3	0.431 ± 1e-2
XGFT-Transformer	0.779 ± 2e-2	0.449 ± 4e-2	0.878 ± 2e-2	0.613 ± 3e-2	**0.783** ± **3e-3**	**0.432** ± **8e-3**

Table 6.1: Performance Comparison for each model.

Method	German		German (SMOTE)		Taiwan	
	AUC	KS	AUC	KS	AUC	KS
No Feature Tokenization	0.605 ± 2e-2	0.205 ± 4e-2	0.802 ± 2e-2	0.503 ± 3e-2	0.773 ± 1e-2	0.415 ± 2e-2
FT-Transfomer	0.76 ± 3e-2	0.436 ± 6e-2	0.86 ± 3e-2	0.588 ± 5e-2	0.782 ± 6e-3	0.431 ± 1e-2
XGFT-Transformer	**0.779** ± **2e-2**	**0.449** ± **4e-2**	**0.878** ± **2e-2**	**0.613** ± **3e-2**	**0.783** ± **3e-3**	**0.432** ± **8e-3**

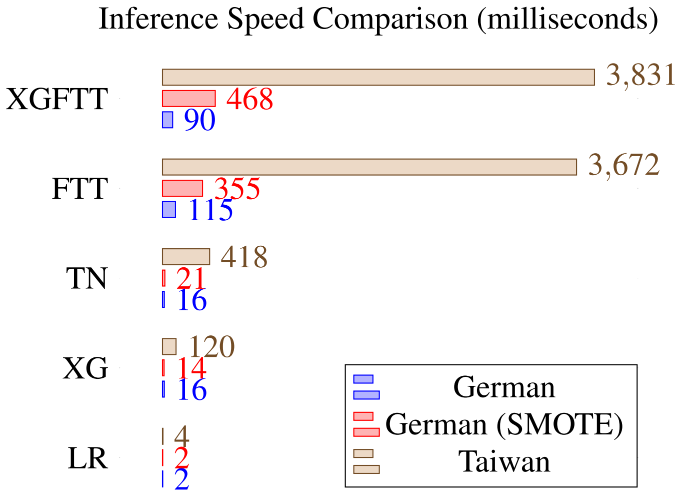
Table 6.2: Ablation Study for the XGFT-Transformer.

The classification AUC and KS scores are displayed in Table 6.1 above, and show an interesting result. XGBoost remains the top performing model on the smaller datasets, however this does not remain the case for the Taiwanese dataset. With access to more data, both the FT-Transformer and XGFT-Transformer surpass its scores, with my novel model achieving the highest performance on the Taiwanese dataset overall. This could be due to the fact it contains a different selection of features, but is more likely due to the large increase in data quantities. Modern deep learning models, especially Transformers, are usually trained on huge datasets and are designed to make use of such quantities of information to learn a superior representation of relationships and trends. Moreover, complex models are known to easily overfit on small datasets, which can be seen most strongly in my experiments with TabNet which performs very poorly on the small German dataset, but almost comparible to XGBoost when given more training data to learn from. The German dataset was chosen for its prevalence in literature, but these results show that it may no longer be a reasonable dataset to use for evaluating modern credit risk models. SMOTE proved to still be an effective method of improving performance on the smaller, imbalanced dataset, despite being a relatively old technique, causing large improvements of up to 0.1 AUC and even greater improvements to KS score. Since KS is a metric that shows how well the model can distinguish binary classes, this shows the minority oversampling successfully assisted the model in learning the minority class. Lastly, by inspecting the confusion matrices, I find that the number of false negatives (defaulters classified as safe) is on average considerably lower for the two Transformer models, which provides further evidence for their ability to distinguish between these difficult samples and handle a minority class well without the need for oversampling.

An ablation study was also performed in Table 5.2 to analyse the effectiveness of different components in my contribution. The feature tokenization module is shown to be a key component that leads to large improvements on tabular data, emphasising the importance of a good input representation. The XGFT-Transformer performs slightly better in both AUC and KS on the Taiwanese dataset, but a larger benefit can be seen in the confidence of results. Across all experiments, the hybrid model notably improves stability and reliability of results, consistently achieving confidence improvements ranging from 1.5 to 2x. This result agrees with my hypothesis, that by allowing the two components to correct for the parts that the other is poor at classifying, the resulting model is more robust.

The results show promise for Transformer-based credit risk models, backing up results from other recent research, with the FT-Transformer showing superior learning ability compared to the other models tested on large enough data. Compared to other similar approaches, Wang et al. [64] also test on the German dataset and achieve superior results, which could be due to multiple factors. Firstly, they use a different Transformer architecture, the SFTransformer, and these results may show that it is more effective on small data. Secondly, they have have simply found a better hyperparameter selection, as there was not enough time in my experiments to perform an exhaustive search. Lastly, this could be a reflection of superior architecture decisions. Unlike the XGFT-Transformer, their model learns in parallel rather than sequentially, and performs a fusion of features to produce the final result, also allowing for a unified loss to be calculated and backpropagated. Of these possibilities, I believe the latter is most likely to lead to noticeable improvements, and is a direction for future work.

6.5.2 Runtime Comparison



One of the tradeoffs of using a larger, more complex model is longer training and inference times, which can be seen in the chart above. The lightweight logistic regression and XGBoost models achieve very fast inference speeds, and for small data this remains under a second on all models. However, as the size of the dataset increases, we see a dramatic increase in inference time for the more complex models. This is an unavoidable effect of larger models, however it does not reach impractical levels and in all experiments inference took no longer than four seconds at most. Also, the

XGFTT model shows only a slight increase in computation time compared to the base FT-Transformer - the majority of computation time is taken up by the FTT half of the model. Further optimization of the Transformer component would be required to allow for even larger datasets, as the trends seen in these results show that with roughly 30x more data, inference takes roughly 30x as long. For a bank with 100,000 or 1,000,000 data points available, inference time would continue to linearly increase following this same trend, reaching impractical levels.

6.6 xAI Evaluation

Method	Dataset	LR	XGB	TabNet	FTT	XGFTT
SHAP	German	33.54	**0.31**	171.58	2842.2	2894.46
GCI	German	**17.96**	24.62	**26.54**	**140.05**	**154.48**
SHAP	Taiwan	1055.67	**8.23**	11121.62	169,520.58	189,720.91
GCI	Taiwan	**601.55**	1054.14	**1661.89**	**8,589.36**	**10,260.77**

Table 6.3: Runtime Comparison in seconds for each global xAI method. Note that TreeSHAP is used for XGBoost, leading to its dramatic speed-up.

To evaluate the effectiveness of Global Counterfactual Importance as an explainability metric I will compare it to SHAP in both efficiency and explanation quality. Firstly, the performance of the algorithms is summarised for both datasets in Table 5.3. With the exception of XGBoost for which TreeSHAP was used, KernelSHAP is consistently far slower than GCI in generating global importance scores. This trend only gets worse for KernelSHAP when the classifier becomes more complex as in the FTT and XGFTT cases, or when the quantity of data to analyse increases, as seen in the Taiwanese dataset where there are 6000 test samples to process. GCI also demonstrates a large increase in computation time, but it is consistently far faster. This is especially noticeable in the case of bigger models - when used to explain the XGFT-Transformer, GCI performs over 18x faster than SHAP. The intensity of calculating Shapley Values, especially for larger data, is one of the known weaknesses of this algorithm. GCI in contrast only has complexity $O(n^2)$ as we just need to iterate over the features of all samples once, and do not consider feature coalitions.

Next, the question of explanation quality is explored - in other words, how accurate are the feature importance scores given by the different algorithms? Since there is no direct evaluation score for explainability, I evaluate the quality of global feature importances by their ranking. Specifically, the top 5 most important features identified by each algorithm are recorded, and then the most important feature that is disagreed on is dropped. The full top 5 for each experiment are recorded in Appendix B. For example, if xAI algorithm A and B both identify feature X as most important, but then identify differing features B and C as second most important, B and C will be dropped for the following experiments. To obtain a numerical evaluation, I compare the performance decrease that occurs when the different features are removed. A greater performance drop means that the feature was indeed important, and the xAI algorithm is performing well. This experiment was performed on both datasets, using Logistic Regression and XGBoost models to allow for quicker runtimes. An immediate discovery from these tests was that for both datasets, SHAP and GCI agree upon the most important feature consistently, but after that disagree on ranks 2-5, etc. As such, Figure 5.2 displays the results of this experiment, with the 'Top Feature' category representing this shared most important feature.

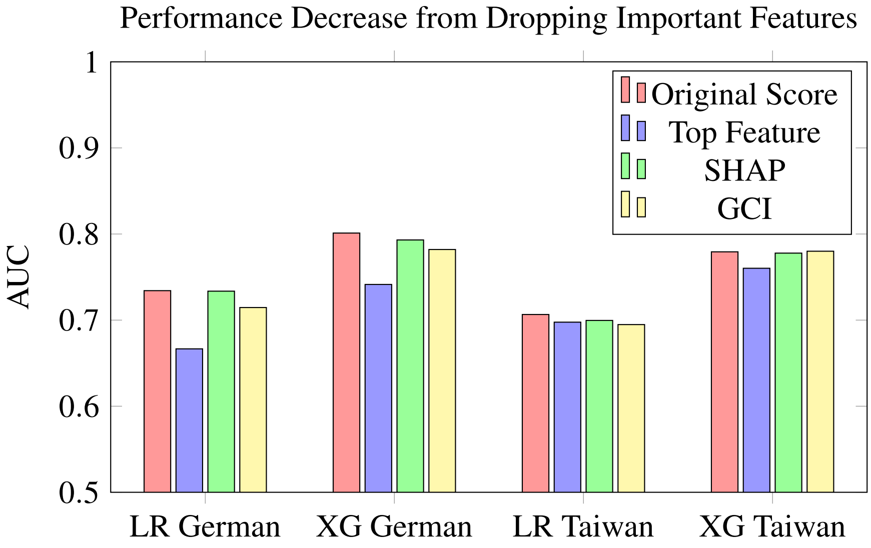


Figure 6.1: Feature importance comparison by removing features deemed important and comparing performance drop.

In all experiments I have found that a single feature dominates the importance scores, but the effects are far more pronounced in the German dataset. 'Status of existing checking account' is always found to be the most important feature in this dataset, which is a reasonable result, and removing this feature causes a considerable loss of roughly 0.07

AUC and 0.13 KS. In the Taiwanese dataset, ‘repayment status in September 2005’ was the dominant feature. The dataset contains multiple variables for repayment status in various months, with September being the final one, alongside bill amounts and amount paid for the same months. The repayment status is a categorical feature representing how many months payment has been delayed for, therefore if there is a clear trend of delayed repayment leading to defaulting, it makes sense for the classifiers to pick up on this. It is a less intuitive feature to be ranked at the top, and this is reflected in the comparatively small performance drop seen in this dataset’s experiments. This smaller drop could be due to the fact the model has been given time series data, and losing one observation month from this is less impactful than losing an entire variable such as account status in the German case.

Since GCI is a less complex algorithm than something like SHAP, it seems logical to expect it to produce lower-quality explanations, however the results seen in Figure 5.2 dispute this. In each scenario, GCI achieves explanations of comparable or superior quality, selecting features in its top 5 that have greater impact on performance than those chosen by SHAP. The exception to this is XGBoost on the Taiwanese dataset, however the difference is only 0.0012 AUC. Outside of the top 5, the algorithms produce much different rankings, and additionally I have found that the results are not deterministic. This applies to both KernelSHAP, in which Shapley Value is estimated to save processing time, and GCI where only one counterfactual per sample is used. This highlights a flaw in the original design, as by only using one counterfactual per sample there is a high variance in values from run to run. On average the top features rank the same each time, but exact values and lower rankings fluctuate due to the high variance. To fix this, multiple counterfactuals would need to be calculated per sample, which would in turn reduce the performance advantage compared to SHAP, since the majority of time is spent calculating the DiCE counterfactuals, not performing GCI itself. Averaging three counterfactuals instead of just using one would triple the overall amount of counterfactuals to be calculated, and with the runtime observations made above it is likely this would roughly triple runtime. For the larger models this is still much faster than KernelSHAP, but for models like LR and XGB it would remove GCI’s advantage.

Chapter 7

Conclusions

7.1 Summary of Results

In this work, I have thoroughly investigated the field of credit risk management and the application of machine learning to this domain. Guided by a comprehensive literature review, I have developed a custom credit risk model, XGFT-Transformer, using a hybrid Transformer architecture, as well as a new, efficient xAI tool, Global Counterfactual Importance (GCI) to answer the lack of explainability seen in similar works. By comparison on two datasets against various competing models, I have demonstrated that while tree-based learning still performs well at a small scale, at a larger scale the expressive power of Transformers leads to superior performance, so long as the data is embedded well. Unlike other works in the area, particularly those that make use of Transformers, XGFTT is fully explainable via any model-agnostic approach, and GCI demonstrates excellent performance in this application compared to competing methods. In summary, by combining tabular transformers with efficient, model-agnostic explainability and harnessing the large quantities of data held by banks, a new state-of-the-art in credit risk can be achieved.

7.2 Future Work

7.2.1 Data Improvements

Our experiments highlight the need for new, larger datasets to support complex models. Tree-based models are only surpassed on larger data, and Transformer-based models like the ones implemented here would likely benefit greatly from a larger quantity of data in the magnitude of $>$100k samples. However, there is some difficulty in obtaining such a large dataset in a publicly available format, as banks which are large enough to have this information are more likely to use it for internal, private research instead.

7.2.2 Model Improvements

The effectiveness of hybrid Transformer-based credit risk models is undeniable, though there are multiple areas for future exploration in the architecture of the XGFT-Transformer. Firstly, while the feature tokenizer used to encode inputs has produced good results in this project, a comparison of different encoding techniques would allow future iterations more confidence when deciding on the optimal way to represent mixed input data. Second, the choice to train the components of the XGFT-Transformer sequentially and seperately rather than using a unified loss function may have impacted performance and a future implementation using parallel training and a combined loss similar to Wang et al. [64] should be considered. Lastly, any model could be used as the hybrid component, not just XGBoost. A comparison of different hybrid components could be performed to obtain evidence for the effectiveness of XGBoost, CNN or other components.

7.2.3 xAI Improvements

While GCI does produce results much faster than other approaches in its category, it can likely be further optimized to reduce computation time significantly, especially on larger models where it still requires multiple hours to compute importances for 6000 test samples. The majority of this computation is spent calculating the DiCE counterfactuals, making efficient counterfactual calculation the first area for improvement here. Another benefit of improving the efficiency of counterfactual calculation is that it would allow for > 1 counterfactual to be produced per sample, which could then be averaged to obtain a more robust importance score and avoid the fluctuations seen in my experiments.

Bibliography

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

[3] Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. Xai for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning*, pages 435–451. PMLR, 2022.

[4] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[5] Amir F Atiya. Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on neural networks*, 12(4):929–935, 2001.

[6] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[7] Przemysław Biecek, Marcin Chlebus, Janusz Gajda, Alicja Gosiewska, Anna Kozak, Dominik Ogonowski, Jakub Sztachelski, and Piotr Wojewnik. Enabling machine learning algorithms for credit scoring–explainable artificial intelligence (xai) methods for clear understanding complex predictive models. *arXiv preprint arXiv:2104.06735*, 2021.

[8] Christine Bolton. *Logistic regression and its application in credit scoring*. University of Pretoria (South Africa), 2009.

[9] Ken Brown and Peter Moles. Credit risk management. *K. Brown & P. Moles, Credit Risk Management*, 16, 2014.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] Bruce G Buchanan and Reid G Smith. Fundamentals of expert systems. *Annual review of computer science*, 3(1):23–58, 1988.

[12] Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57(1):203–216, 2021.

[13] Ahmad Chaddad, Jihao Peng, Jian Xu, and Ahmed Bouridane. Survey of explainable ai techniques in healthcare. *Sensors*, 23(2):634, 2023.

[14] Yung-Chia Chang, Kuei-Hu Chang, and Guan-Jhih Wu. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing*, 73:914–920, 2018.

[15] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021.

[16] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[17] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[18] Financial Crisis Inquiry Commission et al. *The Financial Crisis Inquiry report: the final report of the national commission on the causes of the financial and economic crisis in the united states, including dissenting views*. Cosimo, Inc., 2011.

[19] Jonathan N Crook, David B Edelman, and Lyn C Thomas. Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183(3):1447–1465, 2007.

[20] Xolani Dastile and Turgay Celik. Making deep learning-based predictions for credit scoring explainable. *IEEE Access*, 9:50426–50440, 2021.

[21] Jing Duan. Financial system modeling using deep neural networks (dnns) for effective risk assessment and prediction. *Journal of the Franklin Institute*, 356(8):4716–4731, 2019.

[22] Mona A ElBannan. The financial crisis, basel accords and bank regulations: An overview. *International Journal of Accounting and Financial Reporting*, 7(2):225–275, 2017.

[23] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940. PMLR, 2016.

[24] Timo Freiesleben. The intriguing relation between counterfactual explanations and adversarial examples. *Minds and Machines*, 32(1):77–109, 2022.

[25] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[26] Jorge Galindo and Pablo Tamayo. Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. *Computational economics*, 15:107–143, 2000.

[27] Shreyas Gawde, Shruti Patil, Satish Kumar, Pooja Kamat, Ketan Kotecha, and Sultan Alfarhood. Explainable predictive maintenance of rotating machines using lime, shap, pdp, ice. *IEEE Access*, 12:29345–29361, 2024.

[28] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[29] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data?, 2022.

[30] David J Hand. Modelling consumer credit risk. *IMA Journal of Management mathematics*, 12(2):139–155, 2001.

[31] Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017.

[32] Eiji Hotori, Mikael Wendschlag, and Thibaud Giddey. *Formalization of banking supervision: 19th–20th centuries*. Springer Nature, 2022.

[33] Vicneswary Jahmunah, Eddie YK Ng, Ru-San Tan, Shu Lih Oh, and U Rajendra Acharya. Explainable detection of myocardial infarction using deep learning models with grad-cam technique on ecg signals. *Computers in Biology and Medicine*, 146:105550, 2022.

[34] Jochen Kruppa, Alexandra Schwarz, Gerhard Arminger, and Andreas Ziegler. Consumer credit risk: Individual probability estimates using machine learning. *Expert systems with applications*, 40(13):5125–5131, 2013.

[35] Kin Keung Lai, Lean Yu, Ligang Zhou, and Shouyang Wang. Credit risk evaluation with least square support vector machine. In *Rough Sets and Knowledge Technology: First International Conference, RSKT 2006, Chongquing, China, July 24-26, 2006. Proceedings 1*, pages 490–495. Springer, 2006.

[36] Erkki K Laitinen. Predicting a corporate credit analyst's risk estimate by logistic and linear models. *International review of financial analysis*, 8(2):97–121, 1999.

[37] Tobias Leemann, Alina Fastowski, Felix Pfeiffer, and Gjergji Kasneci. Attention mechanisms don't learn additive models: Rethinking feature importance for transformers. *arXiv preprint arXiv:2405.13536*, 2024.

[38] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020.

[39] Ying Li, Xianghong Lin, Xiangwen Wang, Fanqi Shen, and Zuzheng Gong. Credit risk assessment algorithm using deep neural networks with clustering and merging. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 173–176. IEEE, 2017.

[40] Shu Ling Lin. A new two-stage hybrid approach of credit risk in banking industry. *Expert Systems with Applications*, 36(4):8333–8341, 2009.

[41] Zhaowei Liu, Qianyu Fan, Zhi Wang, and Yeming Cai. A novel algorithm for credit default prediction using tabnet. In *2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 24–27. IEEE, 2023.

[42] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[43] Tambiama Madiega. Artificial intelligence act. *European Parliament: European Parliamentary Research Service*, 2021.

[44] Mohammad Mahbobi, Salman Kimiagari, and Marriappan Vasudevan. Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*, 330(1):609–637, 2023.

[45] Milad Malekipirbazari and Vural Aksakalli. Risk assessment in social lending via random forests. *Expert Systems with Applications*, 42(10):4621–4631, 2015.

[46] Branka Hadji Misheva, Joerg Osterrieder, Ali Hirsa, Onkar Kulkarni, and Stephen Fung Lin. Explainable ai in credit risk management. *arXiv preprint arXiv:2103.00949*, 2021.

[47] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[48] MyFICO. How are fico scores calculated? https://www.myfico.com/credit-education/whats-in-your-credit-score, 2024. Accessed: 07-07-24.

[49] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[50] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[51] Wenyu Qiu. Credit risk prediction in an imbalanced social lending environment based on xgboost. In *2019 5th International Conference on Big Data and Information Analytics (BigDIA)*, pages 150–156. IEEE, 2019.

[52] Shishir Rao, Yikuan Li, Rema Ramakrishnan, Abdelaali Hassaine, Dexter Canoy, John Cleland, Thomas Lukasiewicz, Gholamreza Salimi-Khorshidi, and Kazem Rahimi. An explainable transformer-based deep learning model for the prediction of incident heart failure. *ieee journal of biomedical and health informatics*, 26(7):3362–3372, 2022.

[53] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[54] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[55] Edward H Shortliffe, Randall Davis, Stanton G Axline, Bruce G Buchanan, C Cordell Green, and Stanley N Cohen. Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the mycin system. *Computers and biomedical research*, 8(4):303–320, 1975.

[56] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75, 2021.

[57] Jorge Tejero. Unwrapping black box models: a case study in credit risk. *Revista de Estabilidad Financiera/Banco de España, 43 (otoño 2022), p. 91-122*, 2022.

[58] Anton Frederik Thielmann, Arik Reuter, Thomas Kneib, David Rügamer, and Benjamin Säfken. Interpretable additive tabular transformer networks. *Transactions on Machine Learning Research.*

[59] Geoffrey Towell and Jude Shavlik. Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules. *Advances in neural information processing systems*, 4, 1991.

[60] Chih-Fong Tsai and Ming-Lun Chen. Credit rating by hybrid machine learning techniques. *Applied soft computing*, 10(2):374–380, 2010.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[62] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[63] Kui Wang, Meixuan Li, Jingyi Cheng, Xiaomeng Zhou, and Gang Li. Research on personal credit risk evaluation based on xgboost. *Procedia computer science*, 199:1128–1135, 2022.

[64] Mengyuan Wang, Lijian Zhou, Qingyu Meng, Yifan Kong, and Jie Sun. Credit risk prediction network based on semantic feature transformer and cnn. In *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 723–728. IEEE, 2023.

[65] Yongqiao Wang, Shouyang Wang, and Kin Keung Lai. A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, 13(6):820–831, 2005.

[66] David West. Neural network credit scoring models. *Computers & operations research*, 27(11-12):1131–1152, 2000.

[67] Hongjian Zhang and Katsuhiko Ogasawara. Grad-cam-based explainable artificial intelligence related to medical text processing. *Bioengineering*, 10(9):1070, 2023.

[68] Lei Zhang. The evaluation on the credit risk of enterprises with the cnn-lstm-att model. *Computational Intelligence and Neuroscience*, 2022(1):6826573, 2022.

[69] Zhengyuan Zhang and Zhanquan Wang. Research on credit scoring based on transformer-catboost network structure. In *2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 75–79. IEEE, 2022.

Appendix A

Hyperparameter Search Spaces

Hyperparameter	Search Range
C	$1e-4\ldots 1e2$
max_iter	$100\ldots 2500$
penalty	[None, l1, l2]
solver	[liblinear, lbfgs, sag]

Table A.1: Hyperparameter search space for logistic regression.

Hyperparameter	Search Range
n_estimators	$100\ldots 2000$
max_depth	$2\ldots 10$
learning_rate	$1e-4\ldots 1e-1$
subsample	$0.5\ldots 1$
colsample_bytree	$0.5\ldots 1$
reg_alpha	$0\ldots 10$
reg_lambda	$0\ldots 10$
gamma	$0\ldots 10$

Table A.2: Hyperparameter search space for XGBoost.

Hyperparameter	Search Range
n_d	$2 \dots 64$
n_a	$2 \dots 64$
n_steps	$2 \dots 10$
n_independent	$1 \dots 5$
n_shared	$1 \dots 5$
gamma	$1 \dots 2$
lr	$1e-4 \dots 1e-2$
step_size	$1 \dots 15$
scheduler_gamma	$0.9 \dots 0.999$
weight_decay	$1e-4 \dots 1e-1$
lambda_sparse	$1e-4 \dots 1e-1$

Table A.3: Hyperparameter search space for TabNet.

Hyperparameter	Search Range
input_embed_dim	$8 \dots 64$
num_heads	$4 \dots 12$
num_attn_blocks	$2 \dots 10$
attn_dropout	$0 \dots 0.15$
add_norm_dropout	$0 \dots 0.15$
ff_dropout	$0 \dots 0.15$
learning_rate	$1e-4 \dots 5e-3$
lr_scheduler	[None, LinearLR]

Table A.4: Hyperparameter search space for FT-Transformer.

Appendix B

Top 5 Feature Rankings

Rank	German LR	German XG	Taiwan LR	Taiwan XG
1	Checking Account Status	Checking Account Status	BILL_AMT_AUG	PAY_STATUS_SEP
2	Duration (Months)	Duration (Months)	BILL_AMT_SEP	LIMIT_BAL
3	Age (Years)	Credit History	PAY_STATUS_MAY	AMT_PAID_AUG
4	Credit History	Purpose	LIMIT_BAL	PAY_STATUS_AUG
5	Credit Amount	Credit Amount	PAY_STATUS_JUL	BILL_AMT_SEP

Table B.1: Top 5 Features identified by SHAP

Rank	German LR	German XG	Taiwan LR	Taiwan XG
1	Checking Account Status	Checking Account Status	BILL_AMT_AUG	PAY_STATUS_SEP
2	Duration (Months)	Duration (Months)	PAY_STATUS_MAY	PAY_STATUS_JUN
3	Credit History	Credit History	PAY_STATUS_JUL	BILL_AMT_JUL
4	Credit Amount	Credit Amount	PAY_STATUS_JUN	PAY_STATUS_MAY
5	Age (Years)	Purpose	BILL_AMT_MAY	BILL_AMT_AUG

Table B.2: Top 5 Features identified by GCI