Investigating the Impact of Class Imbalance Handling Methods on Deep Learning with Tabular Data

Master Thesis

presented by
Eun Mi Joo
Matriculation Number 1873015

submitted to the
Data and Web Science Group
Prof. Dr. Heiner Stuckenschmidt
University of Mannheim

July 2024

This thesis explores the impact of various class imbalance handling methods on the performance of deep learning (DL) models applied to tabular data. A comprehensive empirical comparison of different techniques, including data-level approaches and algorithms-level approaches is conducted to identify the most effective strategies on DL models. Additionally, compare between DL and ML performance improvements across different class imbalance handling methods and datasets.

The study provides practical guidance for selecting suitable class imbalance handling methods, emphasizing the importance of techniques like class weight and ROS in enhancing DL model performance. While XGBoost remains a strong performer, DL models can achieve competitive results with effective imbalance handling. This research addresses a gap in existing literature, stimulating further investigation into DL approaches for tabular data and contributing to advancements in the field. Future research could focus on hybrid models, advanced techniques for handling class imbalance, and exploring more diverse datasets.

Contents

1 Introduction **vii**
- 1.1 Background and Motivation vii
- 1.2 Problem Statement vii
- 1.3 Objectives and Research Questions viii

2 Related Work **x**
- 2.1 Overview of Tabular Data and Its Challenges x
- 2.2 Deep Learning Architectures for Tabular Data xii
- 2.3 Machine Learning vs. Deep Learning for Tabular Data xiii
- 2.4 Class Imbalance in Tabular Data xiv
  - 2.4.1 Impacts of Class Imbalance xiv
  - 2.4.2 Class Imbalance Handling Methods xiv
- 2.5 Comparative Studies and Applications xvii
- 2.6 Synthesis and Identification of Gaps xviii

3 Methodology **xix**
- 3.1 Data Collection and Preprocessing xix
- 3.2 Model Selection and Implementation xxi
  - 3.2.1 Deep Learning Models xxi
  - 3.2.2 Gradient Boosted Decision Trees xxi
- 3.3 Experimental Setup xxii

4 Empirical Comparison **xxiii**
- 4.1 Experimental Results xxiii
  - 4.1.1 Detailed Analysis of FT-Transformer xxiii
  - 4.1.2 Detailed Analysis of ResNet xxix
  - 4.1.3 Detailed Analysis of XGBoost xxxiii

5 Discussion **xl**
5.1 Impact of Different Class Imbalance Handling Methods on Deep Learning Models xl
5.2 Comparative Analysis: FT-Transformer vs. ResNet vs. XGBoost . xliv
5.2.1 Comparative Analysis: Key Insights xlv

6 Conclusion **xlviii**
6.1 Summary xlviii
6.2 Future Work xlix

A Program Code / Resources **lv**

B Datasets **lvi**

C Hyperparameter Space **lvii**
C.1 FT-Transformer Hyperparameter Space lvii
C.2 ResNet Hyperparameter Space lviii
C.3 XGBoost Hyperparameter Space lviii

List of Tables

2.1 Sample Tabular Dataset displaying 5 instances consisting of attributes including ID, Age, Gender, Income, and Class, with Gender as the categorical feature. xi

3.1 Dataset properties including imbalance ratios, number of features, categories, samples, and class distributions in training sets. xx

4.1 Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Minority Class). For each dataset, top result is highlighted in Green. xxiv

4.2 Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for FT Transformer. Negative improvements are highlighted in Red. xxv

4.3 Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Majority Class). For each dataset, top result is highlighted in Green. xxvi

4.4 Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for FT Transformer. Negative improvements are highlighted in Red. xxvii

4.5 Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Macro). For each dataset, top result is highlighted in Green. xxviii

4.6 Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Minority Class). For each dataset, top result is highlighted in Green. xxx

4.7 Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for ResNet. Negative improvements are highlighted in Red. xxxi

4.8 Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Majority Class). For each dataset, top result is highlighted in Green. xxxii
4.9 Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for ResNet. Negative improvements are highlighted in Red. xxxiii
4.10 Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Macro). For each dataset, top result is highlighted in Green. xxxiv
4.11 Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Minority Class). For each dataset, top result is highlighted in Green. xxxv
4.12 Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for XGBoost. Negative improvements are highlighted in Red. xxxvi
4.13 Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Majority Class). For each dataset, top result is highlighted in Green. xxxvii
4.14 Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for XGBoost. Negative improvements are highlighted in Red. xxxviii
4.15 Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Macro). For each dataset, top result is highlighted in Green. xxxix

5.1 Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in FT-Transformer. For each dataset, top 3 rank methods are highlighted in Green. xli
5.2 Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in ResNet. For each dataset, top 3 rank methods are highlighted in Green. xli
5.3 Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in XGBoost. For each dataset, top 3 rank methods are highlighted in Green. xlii
5.4 Comparison the Best F1 Score of Models by Class Imbalance Handling Methods and XGBoost's Baseline F1 Score without Any Class Imbalance Handling Methods Across Different Datasets and Imbalance Ratios (IR). For each dataset, top results between DL models (FT-Transformer and ResNet) are in Green xlv

5.5 Subtraction of FT-Transformer - ResNet in F1 Score for Minority Class, Positive values highlighted in Green xlvi
5.6 Subtraction of FT-Transformer - XGBoost in F1 Score for Minority Class, Positive values highlighted in Green xlvi
5.7 Subtraction of ResNet - XGBoost in F1 Score for Minority Class, Positive values highlighted in Green xlvii

C.1 FT-Transformer Hyperparameter space. (A) = ISCXIDS2012, (B) = The rest of the datasets. Use default values for feature embedding size, residual dropout, attention dropout, FFN dropout, and FFN factor, which are determined based on the number of layers. . . . lvii
C.2 ResNet Hyperparameter Space. (A) = ISCXIDS2012, (B) = The rest of the datasets lviii
C.3 XGBoost Hyperparameter Space. lviii

Chapter 1

Introduction

1.1 Background and Motivation

In recent decades, machine learning (ML) and deep learning (DL) have experienced significant advancements, particularly in domains such as computer vision (CV) and natural language processing (NLP). These successes are largely attributed to the development of sophisticated neural network architectures. However, when it comes to tabular data—the oldest and most prevalent form of data in real-world applications—the performance improvements of DL methods have not been as pronounced. Traditional ML approaches, particularly Gradient Boosted Decision Trees (GBDTs) [5, 30], have consistently outperformed DL models for tabular data tasks and remain the preferred choice in many practical scenarios.

Despite the slower progress of DL methods in this area, their flexibility, capability for synthetic data generation, and potential for multimodal integration present compelling reasons to further explore DL approaches for tabular data [2]. These advantages justify a deeper examination of DL for tabular data, particularly in addressing challenges such as class imbalance.

1.2 Problem Statement

The performance gap between ML and DL for tabular data necessitates a thorough examination of the strengths and limitations of both methodologies. Numerous studies [11, 12, 26] have compared the efficacy of DL and traditional ML methods in solving tabular data tasks, aiming to provide comprehensive performance evaluations and propose experimental benchmark environments for fair comparisons. These studies seek to uncover the underlying reasons behind the superior performance of GBDTs and identify scenarios where DL approaches may excel over ML

approaches, thus providing insights to guide future advancements in DL techniques for tabular data.

Despite these efforts, a notable limitation of previous studies is that they often overlook the importance of data preprocessing and inherent data issues such as class imbalance [15, 45], categorical embedding [31], feature engineering [43], and missing values [44]. These issues are frequently tackled using a generalized and unified data preprocessing pipeline across different datasets and models to simplify experiments and focus on model comparison. This approach hinders robust comparisons across different methods and datasets, particularly for tabular data, where effective preprocessing can have a significant impact on model performance.

To build on existing research, data-related issues should be considered thoroughly. Specifically for tabular data, effective data preprocessing and addressing inherent data issues drastically influence model performance. Understanding the role and effect of different data preprocessing and issue-handling methods could bring meaningful advancements in the field of tabular data with DL models. For instance, properly resolving the impact of class imbalance on tabular data is essential in many practical applications.

Class imbalance, a common issue in real-world datasets, occurs when the distribution of samples across different classes is uneven. This imbalance can severely bias the model's predictions, particularly affecting the minority classes. In domains such as credit card fraud detection, disease diagnosis, and cybersecurity, accurate detection of minority classes is critical due to the significant consequences of misclassification. While existing research [4, 6, 14, 15, 25, 45] has extensively investigated algorithms for resolving class imbalance in tabular data, these studies have predominantly focused on ML approaches, including k-nearest neighbors (KNN) [28], logistic regression, decision trees [7], random forests [20], and GBDTs [10]. Although these studies have provided valuable insights, their findings are not directly transferable to DL models. Furthermore, prior research addressing class imbalance in DL has typically focused on text or image data in fields such as NLP [16] and CV [3, 23, 32], rather than tabular data. It still does not fully explain the understanding of DL models when applying different class imbalance handling methods, particularly for tabular data. Consequently, there is a pressing need to fill this gap by examining the influence of class imbalance handling methods on the performance of DL approaches for tabular data.

1.3 Objectives and Research Questions

This thesis aims to explore the impact of various class imbalance handling methods on the performance of DL models when applied to tabular data. The primary

objectives are to compare the effectiveness of different class imbalance handling techniques on DL models, evaluate performance improvements across different DL architectures (e.g., ResNet, Transformer) when these methods are applied, and identify scenarios where DL approaches may outperform traditional ML methods, specifically in the context of class imbalance. Key research questions include how different class imbalance handling methods affect DL models' performance on tabular data, whether certain DL architectures benefit more from these handling methods, and if DL models, with appropriate class imbalance handling, can achieve performance parity or superiority over traditional ML models in tabular data tasks.

To the best of my knowledge, this is the first comprehensive comparison of the performance of various class imbalance handling methods applied to DL models on tabular data by assessing the differences between pre- and post-performance. It compares the performance improvements of different class imbalance handling methods across models and evaluates the performance of different DL models under class imbalance scenarios. Additionally, it compares the performance of DL models with that of the ML method, XGBoost, in tackling class imbalance.

The paper comprises a review of ML and DL approaches for tabular data. It includes an overview of tabular data characteristics, challenges, and a comparison of ML and DL methods. Additionally, the paper reviews different methods for reducing the impact of class imbalance on model performance. The methodology covers data collection, model selection, and experimental setups for fair comparisons between different imbalance handling methods. Experimental results detail the performance of FT-Transformer, ResNet, and XGBoost across various datasets and imbalance ratios. The discussion highlights the impact of these methods on DL models, and the conclusion summarizes key findings and contributions, emphasizing the importance of appropriate imbalance handling.

Chapter 2

Related Work

Despite significant developments in DL in domains like CV and NLP, DL has not yet reached parity with ML methods in tabular data applications. Traditional ML methods, particularly tree-based approaches like GBDTs, consistently outperform DL models in this domain. This disparity arises from several inherent characteristics of tabular data. In this chapter, I discuss the characteristics of tabular data and the challenging properties of tabular data for DL models. I provide a broader overview of existing DL models for tabular data. Additionally, I delve into previous comparative studies between ML and DL on tabular data.

2.1 Overview of Tabular Data and Its Challenges

Tabular data, often referred to as structured data, is the most common data format in practical domains such as finance, healthcare, and manufacturing. It consists of rows and columns, where each row represents an instance of a record, and each column corresponds to an attribute of the respective instance. This format is heterogeneous, containing different types of feature values such as numerical and categorical features. Categorical features are qualitative data represented in numeric or text values without inherent numerical order, indicating descriptive groups. For example, it includes gender, movie genre, and grade.

Tabular data often presents challenges such as missing values and class imbalance. While these issues can challenge both ML and DL models, tree-based approaches like XGBoost can handle them more effectively. For instance, XGBoost can learn a default direction for classifying instances when the splitting feature value is missing [5]. Previous empirical results also support that GBDTs handle skewed distributions better [26]. In contrast, DL models require extensive preprocessing to address these issues, which may potentially introduce noise or result in

information loss.

The example of tabular data can be found in Table 2.1.

Table 2.1: Sample Tabular Dataset displaying 5 instances consisting of attributes including ID, Age, Gender, Income, and Class, with Gender as the categorical feature.

ID	**Age**	**Gender**	**Income ($)**	**Class**
1	25	Male	50000	0
2	30	Female	60000	1
3	22	Female	52000	0
4	35	Male	58000	1
5	28	Female	62000	1

Challenges for DL Models In addition to the previously discussed issues, there are several more challenges for DL models in the tabular domain.

- **Heterogeneity**: Unlike image and text data, which have regular spatial or sequential structures, tabular data consists of heterogeneous attributes with complex and irregular dependencies. This heterogeneity makes it difficult for DL models, which often rely on inductive biases like convolutional neural networks (CNNs) for images or recurrent neural networks (RNNs) for text, to generalize well across different tabular datasets.

- **Dependency on Preprocessing**: The performance of DL models, which do not inherently handle the data issues of tabular data well, is highly dependent on appropriate preprocessing steps. Adequately preprocessing tabular data is challenging and requires domain-specific knowledge to handle properly. Cursory treatment can lead to significant information loss or the introduction of unnecessary noise, further complicating the training process.

- **Feature Importance**: The significance of a single feature value can drastically impact model predictions. Traditional ML approaches, particularly decision tree-based architectures, can inherently handle feature selection and ignore less informative features, whereas DL models lack this capability and require explicit feature engineering.

- **Low Data Quality**: As discussed earlier, tabular datasets frequently suffer from issues such as missing values and class imbalance. While traditional methods like decision tree-based architectures handle these issues robustly,

DL models require careful preprocessing or data handling methods such as missing value imputation, sampling methods to overcome class imbalance, or cost-sensitive learning to resolve class imbalance.

- **Lack of Interpretability**: Compared to decision trees and tree-based models, DL models often lack interpretability of model outcomes. In many applications that use tabular data, such as disease diagnosis in healthcare and credit scoring in finance, it is crucial for practitioners to understand the reasons behind model decisions.

2.2 Deep Learning Architectures for Tabular Data

Recent research has focused on developing and evaluating DL architectures specifically designed for tabular data. Two comprehensive studies [2, 11] provide an in-depth comparison of different ML and DL approaches using several datasets from various domains, such as finance and e-commerce, encompassing different types of tasks like regression and classification. The findings from both studies highlight that popular GBDTs, such as XGBoost, LightGBM, and CatBoost, consistently outperform DL models on most experimental datasets. This superiority holds even when accounting for the longer training times typically required for DL models. Despite extensive research in DL, tree-ensemble methods remain the leading approach for tabular data tasks. However, DL models may surpass traditional ML models on very large datasets, indicating their potential applicability in specific contexts. Effective preprocessing is crucial for enhancing the performance of DL models on tabular data. For instance, transforming the inherent heterogeneity of tabular data into a homogeneous format suitable for DL architectures incurs only a minor additional overhead but significantly improves performance. Standardized preprocessing steps, such as handling categorical features, managing missing values, and applying appropriate scaling, are essential. Additionally, transformer-based architectures have shown promise, offering advantages such as inherent explainability of results and superior performance compared to standard DL architectures. The FT-Transformer [11] has demonstrated strong performance across various datasets, highlighting the effectiveness of transformer-based approaches for tabular data. Similarly, well-tuned ResNet-like architectures achieve competitive performance levels, offering a lightweight and effective baseline for tabular data problems.

These findings suggest that while GBDTs remain the state-of-the-art for most tabular data tasks, advancements in DL, particularly with transformer-based architectures and effective preprocessing techniques, show promising potential. Ongoing research and development in DL methods tailored for tabular data could

further improve performance and expand the applicability of DL in this domain. Consequently, based on the suggestion of the previous research, both ResNet-like and FT-Transformer architectures are selected to be employed as primary DL approaches in this thesis to evaluate their effectiveness and identify scenarios where DL models can outperform traditional ML methods for tabular data.

2.3 Machine Learning vs. Deep Learning for Tabular Data

Shwartz-Ziv and Armon (2021) [37] conducted a study comparing four DL models—NODE [29], DFN-Net [19], TabNet [1], and 1D-CNN—with XGBoost across several datasets. Their findings indicate that while DL models performed well on their original datasets, they did not generalize effectively to other datasets, highlighting their lack of robustness. XGBoost consistently outperformed the DL models in terms of accuracy and computational efficiency, requiring less hyperparameter tuning and offering more reliable performance across different datasets. Notably, combining DL models with XGBoost in an ensemble produced the best results, outperforming both an ensemble of DL models and an ensemble of traditional methods. This suggests that while DL models may struggle independently, they can still provide valuable insights when used in conjunction with tree-based models. These findings highlight the current limitations of DL for tabular data and suggest the potential benefits of hybrid models.

Grinsztajn et al. (2022) [12] investigated why neural networks (NNs) fail to outperform tree-based models, even with extensive hyperparameter tuning. Using 45 carefully selected tabular datasets, the researchers addressed common issues such as missing values and class imbalance through uniform preprocessing steps. Despite comprehensive hyperparameter tuning, NNs did not surpass tree-based models, underscoring the inherent challenges NNs face with tabular data. NN architectures, such as ResNet and MLP, struggled more with learning irregular patterns and were more sensitive to uninformative features compared to tree-based models and Transformer architectures. This suggests a need for better regularization techniques in DL models for tabular data. This aligns with conclusions from other studies [2, 17, 21, 34], which indicate that proper regularization may help NNs learn irregular target functions more effectively. Additionally, the increased sensitivity of standard NN models to uninformative features in tabular data contributes to their poorer performance compared to Transformer-based architectures and tree-based models.

McElfresh et al. (2023) [26] explored the comparative performance of NNs and GBDTs across diverse datasets, aiming to identify the factors that make certain datasets more suitable for NNs. The study suggests that the choice between

DL and ML is often overstated, and recommends light hyperparameter tuning for GBDTs and lightweight DL models like ResNet for meaningful performance gains. The research also highlighted that GBDTs are better suited for datasets with high irregularity and skewed distributions, while NNs perform better on more regular datasets.

2.4 Class Imbalance in Tabular Data

Class imbalance is a common problem in many real-world datasets where the number of instances in one class is significantly larger than in others. This imbalance can distort the learning process for both ML and DL models, which typically assume balanced class distributions. A critical issue arising from class imbalance is that classifiers tend to favor the majority class, as they do not learn sufficiently about the minority classes because of the design structure of the loss function, leading to biased predictions.

2.4.1 Impacts of Class Imbalance

Class imbalance issues frequently appear in real-world tasks where positive cases are rare, such as fraud detection, medical diagnosis, and credit scoring [8, 9, 13, 27, 39, 40]. In these scenarios, identifying the minority class correctly is crucial, and misclassifying minority instances can have severe practical consequences resulting in a high risk. For example, in medical diagnosis, where the minority (positive) class might represent a disease (e.g., cancer), a classifier biased towards the majority (negative) class (e.g., healthy patients) might fail to identify sick patients, leading to detrimental outcomes and high risks for the misclassification of the results.

The main challenge posed by class imbalance is that the training process naturally focuses more on the majority class. During training, the model parameters are updated based on the loss, which is predominantly influenced by the majority class instances. This imbalance skews the learning process, reducing the model's ability to recognize and predict minority class instances accurately.

2.4.2 Class Imbalance Handling Methods

Recognizing the importance of addressing class imbalance in tabular data, numerous studies have proposed advanced methods for improving model performance on class-imbalanced tabular data and compared different class imbalance recovery algorithms across various applications. In this section, I organize different types

of methods for alleviating class imbalance problems in tabular data. These methods can be categorized into data-level approaches, algorithm-level approaches, and hybrid approaches that combine both data and algorithm-level techniques.

Data-Level Approaches

Data-level approaches modify the training dataset to balance the class distribution. This can be achieved through various sampling techniques or by generating synthetic data.

- **Over-Sampling** Over-sampling techniques increase the number of minority class instances to match the majority class, preventing the model from being biased towards the majority class and enhancing its ability to correctly classify minority samples. Consequently, these techniques improve the classifier's performance in accurately identifying both majority and minority classes, leading to a more balanced and effective model.
  - Random Over-Sampling (ROS): This method randomly duplicates minority class instances to balance class distribution in the dataset. Due to its simplicity and effectiveness, it is a popular choice among practitioners for addressing class imbalance issues. However, it may potentially lead to overfitting on minority samples, as the model may fit too closely to the duplicated instances, reducing its generalizability.
  - Synthetic Minority Over-sampling Technique (SMOTE): SMOTE [4] generates synthetic samples by interpolating between existing minority instances and randomly chosen instances among their k-nearest neighbors. This strategy can help overcome the problem of overfitting. However, interpolation-based sample generation can potentially lead to a noisy dataset.
  - Adaptive Synthetic Sampling Approach (ADASYN): Similar to SMOTE, ADASYN [14] generates synthetic samples for minority instances, but it focuses on the minority samples that are more challenging for the model to learn, identified by their proximity to majority class instances, thus focusing on minority instances close to the boundary. Additionally, it adaptively decides how many instances to generate based on the density of the data distribution.
  - Generative Adversarial Networks (GANs): GANs [33] generate synthetic instances through a generator-discriminator model architecture. Although primarily used for image and text data, GANs can be adapted for tabular data. They can generate synthetic instances more flexibly

compared to SMOTE and ADASYN. However, they require significant computational resources and need to be trained for each dataset.

- **Under-Sampling** Under-sampling techniques reduce the number of majority class instances to create a more balanced dataset.
  - Random Under-Sampling (RUS): This method randomly samples and removes instances from the majority class to balance class distribution in the dataset. Similar to ROS, due to its simplicity, it is frequently used by practitioners when dealing with class-imbalanced datasets. However, this approach can lead to the loss of important information as potentially valuable majority class instances may be discarded.
  - Tomek Links: This method removes instances that form Tomek links, which are pairs of samples from different classes that are each other's nearest neighbors, considered as noise.
  - Edited Nearest Neighbors (ENN): The purpose of ENN [42] is similar to that of Tomek Links, but the selection of the noise is different. ENN removes instances where the majority class of the k-nearest neighbors differs from the instance's class.
  - Repeated ENN (RENN): RENN is a variation of ENN, which repeatedly applies ENN to further clean the dataset.
  - All KNN: A variation of RENN, it also repeatedly applies ENN, but during the iteration, it increases the value of k in each iteration.
- **Adjusting Batch Strategies** This approach [35] ensures that each mini-batch used in training contains a balanced number of samples from each class, samplings for the mini-batch include oversampling minority samples and undersampling majority samples for the batch.

Algorithm-Level Approaches

Algorithm-level approaches adjust the learning process to account for class imbalance instead of directly altering the data itself. These methods include cost-sensitive learning.

- **Cost-Sensitive Learning** This approach assigns different misclassification costs to instances depending on the type of class. In class imbalance circumstances, this ensures that misclassifications of minority instances have a higher penalty. One frequently used practice is adopting a cost-sensitive loss function in the training process. It directly incorporates class-specific costs

into the loss function to penalize misclassification of minority instances more heavily.

Hybrid Approaches

- **Two-Phase Training** This algorithm [24] involves initially training a model on a balanced dataset using sampling techniques such as ROS and RUS and then fine-tuning the classification layers on the original imbalanced dataset. This effectively improves minority class performance with minimal impact on majority class accuracy.

2.5 Comparative Studies and Applications

Dey et al. (2023) [6] compare different methods for addressing class imbalance problems on tabular data: SMOTE, Borderline-SMOTE, and ADASYN, using classifiers such as SVM, KNN, GaussianNB, Decision Tree, and Random Forest. The evaluation metrics include F1-score, AUC ROC, accuracy, precision, and recall. The results indicate no single superior method among the three. Recognizing that the models used in this study are limited to traditional ML classifiers, it highlights the need for further exploration with advanced classifiers like DL models and GBDTs. Buda et al. (2017) [3] investigate the impact of class imbalance on CNN performance using datasets like MNIST, CIFAR-10, and ImageNet. The study highlights the importance of addressing class imbalance in DL tasks and provides a comprehensive evaluation of methods to tackle it. It compares oversampling, undersampling, thresholding, and two-phase training, finding that oversampling consistently outperforms other methods without causing overfitting. Henning and Rehbein (2022) [16] highlight the importance of addressing class imbalance in NLP and provide a structured overview of current approaches. This survey categorizes class imbalance into step imbalance, linear imbalance, and long-tailed distribution based on the form of the distribution of the class imbalance. The survey provides a fair experimental framework for comparing different methods to address class imbalance across different datasets. For instance, both controlled distribution datasets and real-world datasets are used. Synthetically controlled experimental factors (e.g., imbalance ratio) help researchers investigate several factors at once. Additionally, the effectiveness of the methods on real-world practices is also evaluated on naturally imbalanced data. They review methods such as re-sampling, data augmentation, loss function adjustments, staged learning, and model design modifications. The study demonstrates that re-sampling, data augmentation, and changing the loss function are straightforward choices, while staged learning and

model design modifications offer more advanced but potentially higher-cost solutions.

2.6 Synthesis and Identification of Gaps

Integrating these findings highlights several gaps in the existing research. Despite extensive work on class imbalance handling methods in traditional ML, there is limited research on their application to DL models, particularly for tabular data. Additionally, while several studies have evaluated different DL architectures for tabular data, a systematic comparison of the impact of various class imbalance handling methods across these architectures has not been explored. This thesis aims to fill these gaps by providing a comprehensive empirical comparison of class imbalance handling methods applied to various DL models for tabular data and a comparison between different models, including DL and ML. Furthermore, the study seeks to determine if there is a dominant method to alleviate class imbalance for DL models. Finally, it aims to identify any scenarios where DL models can outperform traditional ML models after addressing class imbalance in both cases.

Chapter 3

Methodology

This study aims to investigate the impact of various class imbalance handling methods on the performance of DL models when applied to imbalanced tabular data. The research involves a comparative analysis between XGBoost for ML approaches, and ResNet and FT-Transformer for DL approaches under different class imbalance scenarios. The primary focus is on evaluating the effectiveness of each method in improving the performance of DL models on imbalanced tabular datasets.

3.1 Data Collection and Preprocessing

The datasets used in this study were sourced from various domains, including finance and cybersecurity for real-world data, which commonly encounter class imbalance issues. Each dataset underwent the following preprocessing steps.

- **Data Cleaning**: Missing values were imputed with zeros to maintain data integrity.
- **Feature Transformation**: Categorical features were transformed using one-hot encoding. For FT-Transformer, categorical features were explicitly specified due to its distinct internal handling of different feature types.
- **Data Splitting**: The datasets were split into training, validation, and test sets, ensuring that the same test sets were used across all algorithms for robust evaluation. The test set splitting ensured equal representation of minority and majority classes (50:50 ratio). This balanced test set allows for accurate evaluation of the performance of each combination of class imbalance handling methods and models in terms of F1 score. This approach follows the

findings of Williams et al. (2020) [41], which highlight the precision-recall curve's vulnerability to imbalanced test sets.

  - Synthetically Imbalanced Dataset (Controlled Dataset): Two datasets were selected from the study [12] provided by OpenML for generating controlled class imbalance scenarios, using imbalance ratios (IRs) of 100, 20, 10, and 2. Balanced datasets (IR of 1) were also generated for baseline comparisons, to understand how difficult this dataset is for the models without any class imbalance influence in the model training. The imbalance was generated after the train-test dataset split to ensure a sufficient number of minority samples is designated for the test set as well.
  - Real-World Dataset: Two real-world datasets were used to assess performance in practical class imbalance scenarios.
    * Give Me Some Credit: The GiveMeSomeCredit dataset contains borrowers' financial history and demographic data for credit risk prediction [18].
    * Intrusion Detection Evaluation Dataset (ISCXIDS2012): The ISCXIDS2012 dataset, which includes network traffic data capturing various types of network behavior, including normal activities and cyber-attacks, is used to assess performance in practical class imbalance scenarios [36].

- **Cross-Validation**: For model selection, 3-fold cross-validation with 15 sets of hyperparameters was used. The training-validation sets were stratified, and the mean and standard deviation of the F1-score were reported as the model performance evaluation metric.

The controlled and real-world datasets' properties are summarized in Table 3.1.

Table 3.1: Dataset properties including imbalance ratios, number of features, categories, samples, and class distributions in training sets.

Dataset	electricity					MagicTelescope					GMSC	ISCXIDS2012
IR	100	20	10	2	1	100	20	10	2	1	17	55
# of features	8	8	8	8	8	10	10	10	10	10	11	16
# of categorical features	1	1	1	1	1	0	0	0	0	0	0	3
# of samples in test set (majority + minority)	10430	10430	10430	10430	10430	4932	4932	4932	4932	4932	4010	1510
# of majority samples in train set	20860	20860	20860	20860	10430	9866	9866	9866	8444	4222	137969	166849
# of minority samples in train set	208	1043	2086	10430	10430	98	493	986	4222	4222	8021	3021

3.2 Model Selection and Implementation

Two main types of DL models were used in the experiments,: ResNet and FT-Transformer, and XGBoost for ML as a baseline model.

3.2.1 Deep Learning Models

The implementation of ResNet and FT-Transformer by [11] was used for the experiments.

- **ResNet**: The ResNet architecture is designed to address gradient vanishing problems by adding residual connections that skip layers, allowing inputs to directly influence subsequent layers. This study [11] used a simplified version of ResNet for better optimization.
- **FT-Transformer (Feature Tokenizer + Transformer)**: This model adapts the Transformer architecture for tabular data, using a feature tokenizer to transform both numerical and categorical features into embeddings. These embeddings are then processed by a stack of Transformer layers to obtain the final representation.

For hyperparameter optimization, Weights & Biases package was used for the hyperparameter searching process. Bayesian optimization was used to find the optimal hyperparameters for the models. The hyperparameter search involved 15 iterations for each dataset and 30 iterations for DL models on the ISCXIDS2012 dataset. The best models were selected based on validation set performance. The `AdamW` optimizer was used for DL models without learning rate schedules. This experiment is limited to binary classification tasks. Therefore, binary cross-entropy (BCE) was used for the loss function, and weighted BCE for class-weight method to address class imbalance. Learning rate, weight decay, and model architecture parameters were included in the hyperparameter search. Details of the hyperparameter search spaces used for each model are provided in Appendix C. Early stopping with a patience of 10 epochs was used for both ResNet and FT-Transformer.

3.2.2 Gradient Boosted Decision Trees

As a baseline model, XGBoost was chosen due to its popularity for tabular data. The same cross-validation procedure was applied as in the DL training process. Bayesian optimization was also used for XGBoost. The `BayesianOptimization` package was utilized. The hyperparameter search iteration count was fixed to 15 across all datasets in the experiment.

3.3 Experimental Setup

The experiments were designed to evaluate the impact of different class imbalance handling methods on model performance. The methods tested include,

:

- Over-Sampling: Techniques such as ROS, SMOTE, and ADASYN.
- Under-Sampling: Methods like RUS and noise reduction techniques such as ENN, RENN, and All KNN.
- Cost-Sensitive Loss Function (Class Weight): Adjusting loss functions to assign higher penalties for misclassifying minority class instances, with costs calculated by imbalance ratios of the train datasets.
- Balanced Batch Training: Ensuring each mini-batch contains a balanced number of samples from each class. Both random over-sampling and random under-sampling techniques for sampling mini-batch instances were tested, along with stratified sampled mini-batch training, which ensures that each batch has the same imbalance ratio as the original train dataset.

In previous studies on class imbalance, multiple evaluation metrics such as F1-score, AUC ROC, accuracy, precision, and recall are commonly used. According to the study by Velarde et al. (2023) [38], accuracy can become misleading in scenarios with more severe class imbalance, and the ROC curve, despite its widespread use, is not appropriate for evaluating imbalanced datasets. The study emphasizes that F scores already encapsulate both precision and recall, with variations in F scores reflecting the relative importance of precision versus recall. Consequently, this paper uses the F1-score as the primary evaluation metric, as it balances the importance of both precision and recall equally, providing a comprehensive measure of how well the models handle class imbalance and generalize to unseen data.

Chapter 4

Empirical Comparison

4.1 Experimental Results

In this section, different models are evaluated across various datasets and imbalance ratios to understand their performance in handling class imbalance. The F1 scores for the minority class, majority class, and macro average are measured. Furthermore, the improvements in F1 scores for both the minority and majority classes for each method, compared to the no-treatment baseline score, are calculated by subtracting the F1 scores of each method from the baseline score. This analysis aims to understand how much these techniques mitigate the impact of class imbalance across different models.

4.1.1 Detailed Analysis of FT-Transformer

The F1 scores for the minority class, majority class, and macro average for FT-Transformer are presented in Table 4.1, Table 4.3, and Table 4.5, respectively. Additionally, the improvements in F1 scores for both the minority and majority classes for each method, compared to the no-treatment baseline, are shown in Tables 4.2 and Table 4.4.

Synthetically Imbalanced Dataset with Higher Imbalance Ratio

For datasets with a high imbalance ratio (IR 100 and IR 20), the baseline F1 score without any class imbalance handling is very low, indicating the FT-Transformer's poor performance in handling such datasets. Despite the low baseline score, the class weight method consistently performs best, significantly improving model performance compared to the baseline. Furthermore, This method not only enhances the F1 score for the minority class but also for the majority class. ADASYN

Table 4.1: Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Minority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.009 ± 0.008	0.730 ± 0.016	0.331 ± 0.041	0.418 ± 0.028	0.461 ± 0.014	0.335 ± 0.196	0.026 ± 0.019	0.004 ± 0.003	0.027 ± 0.006	0.375 ± 0.022	0.378 ± 0.026	0.042 ± 0.059
electricity	20	0.205 ± 0.014	0.752 ± 0.012	0.726 ± 0.012	0.738 ± 0.016	0.726 ± 0.008	0.720 ± 0.004	0.377 ± 0.027	0.340 ± 0.033	0.243 ± 0.102	0.747 ± 0.005	0.707 ± 0.044	0.260 ± 0.042
electricity	10	0.485 ± 0.018	0.770 ± 0.013	0.761 ± 0.016	0.768 ± 0.009	0.777 ± 0.009	0.734 ± 0.020	0.478 ± 0.076	0.515 ± 0.071	0.542 ± 0.030	0.760 ± 0.008	0.756 ± 0.017	0.496 ± 0.038
electricity	2	0.789 ± 0.005	0.824 ± 0.014	0.815 ± 0.005	0.825 ± 0.007	0.819 ± 0.003	0.798 ± 0.010	0.781 ± 0.012	0.782 ± 0.003	0.785 ± 0.009	0.816 ± 0.019	0.815 ± 0.005	0.792 ± 0.007
electricity	1	0.805 ± 0.017											
MagicTelescope	100	0.262 ± 0.027	0.741 ± 0.004	0.481 ± 0.011	0.549 ± 0.046	0.533 ± 0.014	0.668 ± 0.018	0.310 ± 0.056	0.268 ± 0.013	0.313 ± 0.035	0.503 ± 0.049	0.477 ± 0.029	0.244 ± 0.038
MagicTelescope	20	0.542 ± 0.037	0.772 ± 0.006	0.705 ± 0.027	0.744 ± 0.012	0.711 ± 0.011	0.757 ± 0.008	0.562 ± 0.044	0.544 ± 0.018	0.548 ± 0.023	0.703 ± 0.022	0.712 ± 0.006	0.547 ± 0.043
MagicTelescope	10	0.633 ± 0.013	0.770 ± 0.003	0.747 ± 0.006	0.756 ± 0.007	0.778 ± 0.010	0.741 ± 0.006	0.643 ± 0.020	0.627 ± 0.009	0.650 ± 0.006	0.785 ± 0.021	0.775 ± 0.013	0.633 ± 0.023
MagicTelescope	2	0.820 ± 0.010	0.822 ± 0.007	0.828 ± 0.008	0.827 ± 0.003	0.828 ± 0.002	0.829 ± 0.002	0.800 ± 0.013	0.789 ± 0.005	0.789 ± 0.004	0.829 ± 0.004	0.825 ± 0.011	0.798 ± 0.019
MagicTelescope	1	0.828 ± 0.004											
GMSC	17	0.242 ± 0.051	0.770 ± 0.003	0.764 ± 0.006	0.739 ± 0.014	0.732 ± 0.012	0.736 ± 0.025	0.240 ± 0.028	0.285 ± 0.036	0.236 ± 0.033	0.758 ± 0.012	0.763 ± 0.009	0.242 ± 0.029
ISCXIDS2012	55	0.000 ± 0.000	0.763 ± 0.005	0.604 ± 0.427	0.607 ± 0.429	0.214 ± 0.301	0.766 ± 0.006	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.886 ± 0.029	0.849 ± 0.066	0.000 ± 0.000

Table 4.2: Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for FT Transformer. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.721	0.322	0.409	0.452	0.326	0.017	-0.005	0.018	0.367	0.370	0.033
electricity	20	0.547	0.521	0.533	0.521	0.515	0.172	0.134	0.037	0.542	0.502	0.055
electricity	10	0.285	0.275	0.283	0.291	0.249	-0.007	0.030	0.057	0.275	0.271	0.010
electricity	2	0.036	0.026	0.036	0.030	0.009	-0.007	-0.007	-0.004	0.028	0.027	0.003
MagicTelescope	100	0.479	0.219	0.287	0.271	0.407	0.048	0.006	0.051	0.241	0.215	-0.018
MagicTelescope	20	0.230	0.163	0.202	0.170	0.215	0.020	0.003	0.006	0.161	0.170	0.005
MagicTelescope	10	0.137	0.114	0.123	0.145	0.108	0.010	-0.006	0.017	0.152	0.142	-0.000
MagicTelescope	2	0.002	0.008	0.008	0.008	0.009	-0.020	-0.031	-0.031	0.010	0.005	-0.022
GMSC	17	0.528	0.522	0.497	0.489	0.494	-0.003	0.043	-0.006	0.515	0.521	-0.001
ISCXIDS2012	55	0.763	0.604	0.607	0.214	0.766	0.000	0.000	0.000	0.886	0.849	0.000

and SMOTE also show substantial improvements among the methods that balance the number of class samples but are less effective than the class weight method. While ROS and RUS indicate that simple adjustments help but are less effective than more sophisticated techniques in electricity dataset. However, in MagicTelescope dataset with a higher imbalance ratio (IR 100 and IR 20), the outperforming method shifts from synthetic sample generation methods (ADASYN and SMOTE) to RUS among the methods that address class imbalance by balancing the number of classes (ROS, ADASYN, SMOTE, and RUS). Batch Balanced Over/Under Sampling shows moderate improvements but is not as effective as the methods that balance class samples. Noise reduction techniques (ENN, RENN, All KNN, and Batch Stratified) show very poor performance across different datasets.

Synthetically Imbalanced Dataset with Lower Imbalance Ratio

As the imbalance ratio decreases (IR 10 and IR 2), the baseline F1 score improves but not as much as in XGBoost. The difference in efficacy between different class imbalance handling methods becomes less significant. While class weight remains a strong performer, ADASYN and SMOTE outperform it in electricity dataset, and Batch Balancing by over-sampling outperforms it in MagicTelescope dataset. Noise reduction techniques continue to underperform. Additionally, achieving a higher F1 score for the minority class by using class imbalance methods often coincides with a decrease in the F1 score for the majority class in both datasets electricity and MagicTelescope.

Real-World Dataset

In the GiveMeSomeCredit dataset, the baseline performance is low, but class weight achieves the best improvement in the F1 score for the minority class without caus-

Table 4.3: Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Majority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.668 ± 0.001	0.739 ± 0.029	0.706 ± 0.007	0.717 ± 0.005	0.722 ± 0.003	0.651 ± 0.013	0.670 ± 0.002	0.667 ± 0.000	0.670 ± 0.001	0.710 ± 0.004	0.708 ± 0.004	0.672 ± 0.007
electricity	20	0.692 ± 0.002	0.777 ± 0.009	0.782 ± 0.001	0.780 ± 0.003	0.778 ± 0.002	0.777 ± 0.003	0.719 ± 0.005	0.712 ± 0.005	0.698 ± 0.014	0.775 ± 0.007	0.776 ± 0.004	0.700 ± 0.006
electricity	10	0.742 ± 0.004	0.779 ± 0.007	0.789 ± 0.003	0.793 ± 0.004	0.792 ± 0.006	0.778 ± 0.005	0.739 ± 0.015	0.745 ± 0.016	0.752 ± 0.007	0.785 ± 0.005	0.782 ± 0.006	0.743 ± 0.008
electricity	2	0.826 ± 0.004	0.825 ± 0.007	0.822 ± 0.008	0.824 ± 0.009	0.808 ± 0.007	0.803 ± 0.005	0.786 ± 0.009	0.744 ± 0.002	0.778 ± 0.007	0.826 ± 0.009	0.819 ± 0.015	0.827 ± 0.005
electricity	1	0.810 ± 0.009											
MagicTelescope	100	0.702 ± 0.004	0.783 ± 0.004	0.740 ± 0.003	0.754 ± 0.011	0.748 ± 0.002	0.765 ± 0.010	0.709 ± 0.010	0.702 ± 0.002	0.710 ± 0.006	0.745 ± 0.011	0.739 ± 0.006	0.698 ± 0.006
MagicTelescope	20	0.759 ± 0.010	0.803 ± 0.005	0.797 ± 0.010	0.806 ± 0.004	0.797 ± 0.006	0.798 ± 0.001	0.764 ± 0.013	0.759 ± 0.005	0.760 ± 0.006	0.796 ± 0.004	0.797 ± 0.004	0.761 ± 0.011
MagicTelescope	10	0.785 ± 0.003	0.806 ± 0.003	0.812 ± 0.000	0.811 ± 0.002	0.812 ± 0.011	0.806 ± 0.003	0.787 ± 0.006	0.782 ± 0.002	0.788 ± 0.001	0.825 ± 0.010	0.810 ± 0.011	0.785 ± 0.007
MagicTelescope	2	0.852 ± 0.002	0.845 ± 0.005	0.853 ± 0.003	0.847 ± 0.003	0.841 ± 0.004	0.840 ± 0.001	0.815 ± 0.014	0.784 ± 0.003	0.802 ± 0.004	0.850 ± 0.003	0.849 ± 0.002	0.845 ± 0.007
MagicTelescope	1	0.845 ± 0.005											
GMSC	17	0.695 ± 0.007	0.769 ± 0.006	0.773 ± 0.006	0.760 ± 0.001	0.764 ± 0.005	0.765 ± 0.007	0.695 ± 0.004	0.700 ± 0.005	0.694 ± 0.004	0.774 ± 0.003	0.774 ± 0.006	0.695 ± 0.004
ISCXIDS2012	55	0.667 ± 0.000	0.588 ± 0.015	0.818 ± 0.107	0.821 ± 0.109	0.685 ± 0.037	0.599 ± 0.020	0.667 ± 0.000	0.667 ± 0.000	0.667 ± 0.000	0.859 ± 0.047	0.774 ± 0.147	0.667 ± 0.000

Table 4.4: Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for FT Transformer. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.071	0.038	0.049	0.054	-0.016	0.002	-0.001	0.002	0.042	0.040	0.004
electricity	20	0.085	0.090	0.088	0.086	0.085	0.027	0.021	0.006	0.083	0.084	0.008
electricity	10	0.038	0.047	0.052	0.051	0.036	-0.003	0.004	0.010	0.043	0.040	0.002
electricity	2	-0.001	-0.004	-0.002	-0.019	-0.024	-0.040	-0.082	-0.048	-0.001	-0.007	0.001
MagicTelescope	100	0.081	0.039	0.053	0.046	0.063	0.008	0.001	0.008	0.044	0.037	-0.003
MagicTelescope	20	0.044	0.038	0.046	0.037	0.039	0.004	-0.001	0.000	0.037	0.037	0.001
MagicTelescope	10	0.021	0.027	0.026	0.027	0.021	0.002	-0.003	0.004	0.040	0.025	-0.000
MagicTelescope	2	-0.008	0.001	-0.006	-0.011	-0.012	-0.037	-0.068	-0.050	-0.003	-0.004	-0.007
GMSC	17	0.075	0.079	0.065	0.069	0.071	-0.000	0.005	-0.001	0.079	0.080	0.000
ISCXIDS2012	55	-0.079	0.151	0.154	0.019	-0.068	0.000	0.000	0.000	0.193	0.107	0.000

ing a decrease in the F1 score for the majority class similar to the result from the controlled class imbalanced datasets. ROS follows in effectiveness. SMOTE, ADASYN, RUS, and Balanced Batch Over/Under Sampling show comparable improvements to the best methods, while noise reduction techniques and Batch Stratified perform the worst, sometimes even detrimentally affecting the F1 score. In the ISCXIDS2012 dataset, because of a higher imbalance ratio, the baseline F1 score is 0. Batch Over Sampling and Batch Under Sampling show significant improvements, while class weight, ROS, SMOTE, and RUS provide moderate improvements. ADASYN shows some improvement but is less effective compared to other methods. Noise reduction techniques do not bring any benefits for handling class imbalance in the model training, as validated by F1 scores.

The best methods across various imbalance ratios and datasets for the FT-Transformer are consistently class weight, over-sampling techniques like ADASYN, SMOTE and RUS. Batch balancing methods and ROS provide moderate improvements, while noise reduction techniques (ENN, RENN, All KNN) and Batch Stratified methods often underperform. These performance trends are consistent across controlled datasets (electricity and MagicTelescope) and real-world datasets (GiveMeSomeCredit and ISCXIDS2012), highlighting the broad applicability of these findings. This detailed analysis provides a comprehensive understanding of how different class imbalance handling methods impact model performance across various datasets and imbalance ratios, particularly for the FT-Transformer model.

Regarding both the F1 score for the minority class and the majority class, noise reduction techniques (ENN, RENN, All KNN) consistently show a pattern where a decrease in the F1 score for the minority class is accompanied by a decrease in the F1 score for the majority class. Conversely, when methods yield the best F1 scores for the minority class, in most cases (especially in datasets with higher imbalance ratios such as IR 100 and IR 20, and real-world data), the F1 score for the majority

Table 4.5: Empirical Comparison of Class Imbalance Handling Method for FT Transformer (F1 Score Macro). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.338 ± 0.005	0.734 ± 0.013	0.518 ± 0.024	0.567 ± 0.016	0.591 ± 0.008	0.493 ± 0.092	0.348 ± 0.011	0.335 ± 0.001	0.348 ± 0.003	0.543 ± 0.012	0.543 ± 0.015	0.357 ± 0.033
electricity	20	0.449 ± 0.008	0.765 ± 0.002	0.754 ± 0.006	0.759 ± 0.009	0.752 ± 0.005	0.749 ± 0.002	0.548 ± 0.016	0.526 ± 0.019	0.470 ± 0.058	0.761 ± 0.005	0.741 ± 0.024	0.480 ± 0.024
electricity	10	0.614 ± 0.011	0.775 ± 0.004	0.775 ± 0.009	0.781 ± 0.006	0.785 ± 0.006	0.756 ± 0.007	0.608 ± 0.046	0.630 ± 0.043	0.647 ± 0.018	0.772 ± 0.005	0.769 ± 0.008	0.620 ± 0.023
electricity	2	0.807 ± 0.005	0.825 ± 0.010	0.819 ± 0.003	0.825 ± 0.008	0.813 ± 0.005	0.800 ± 0.007	0.784 ± 0.011	0.763 ± 0.002	0.781 ± 0.007	0.821 ± 0.013	0.817 ± 0.009	0.810 ± 0.006
electricity	1	0.808 ± 0.012											
MagicTelescope	100	0.482 ± 0.015	0.762 ± 0.001	0.611 ± 0.007	0.652 ± 0.028	0.640 ± 0.008	0.717 ± 0.014	0.509 ± 0.033	0.485 ± 0.007	0.511 ± 0.021	0.624 ± 0.030	0.608 ± 0.017	0.471 ± 0.022
MagicTelescope	20	0.651 ± 0.023	0.788 ± 0.005	0.751 ± 0.018	0.775 ± 0.008	0.754 ± 0.008	0.778 ± 0.004	0.663 ± 0.028	0.651 ± 0.011	0.654 ± 0.015	0.749 ± 0.013	0.754 ± 0.004	0.654 ± 0.027
MagicTelescope	10	0.709 ± 0.008	0.788 ± 0.003	0.780 ± 0.003	0.783 ± 0.004	0.795 ± 0.009	0.773 ± 0.003	0.715 ± 0.013	0.705 ± 0.006	0.719 ± 0.004	0.805 ± 0.015	0.793 ± 0.012	0.709 ± 0.015
MagicTelescope	2	0.836 ± 0.006	0.833 ± 0.002	0.841 ± 0.005	0.837 ± 0.001	0.834 ± 0.003	0.835 ± 0.001	0.807 ± 0.013	0.787 ± 0.003	0.795 ± 0.004	0.839 ± 0.003	0.837 ± 0.006	0.821 ± 0.012
MagicTelescope	1	0.837 ± 0.005											
GMSC	17	0.469 ± 0.029	0.770 ± 0.002	0.769 ± 0.003	0.749 ± 0.006	0.748 ± 0.008	0.751 ± 0.010	0.467 ± 0.016	0.493 ± 0.021	0.465 ± 0.019	0.766 ± 0.007	0.769 ± 0.007	0.468 ± 0.016
ISCXIDS2012	55	0.333 ± 0.000	0.675 ± 0.010	0.711 ± 0.267	0.714 ± 0.269	0.450 ± 0.168	0.682 ± 0.013	0.333 ± 0.000	0.333 ± 0.000	0.333 ± 0.000	0.873 ± 0.038	0.812 ± 0.106	0.333 ± 0.000

class also improves compared to the no-treatment baseline score.

4.1.2 Detailed Analysis of ResNet

The F1 scores for the minority class, majority class, and macro average for ResNet are presented in Table 4.6, Table 4.8, and Table 4.10. Similar to the FT-Transformer, the improvements for both F1 scores of the minority class and the majority class for each method are presented in Table 4.7 and Table 4.9.

Synthetically Imbalanced Dataset with Higher Imbalance Ratio

Similar to the results shown in FT-Transformer, for datasets with a high imbalance ratio (IR 100 and IR 20), the baseline F1 score without any class imbalance handling is very low, indicating poor performance of ResNet in handling class imbalance inherent in the model. However, ResNet generally shows greater performance than the FT-Transformer on the baseline datasets, specifically electricity and MagicTelescope, with a high imbalance ratio. Methods that achieve the biggest improvement in the F1 score for the minority class also tend to show significant improvements in the F1 score for the majority class, achieving results similar to those methods that achieve the highest improvements in the F1 score for the majority class. The class weight method is a prevalent approach to handling class imbalance for ResNet. Although its efficacy is not as dominant as it is for the FT-Transformer, RUS and Batch Balancing methods sometimes outperform it. Nonetheless, even when these methods take the lead, the improvement achieved by class weight remains comparable. When class weight is the most effective method, other methods like ROS, RUS, and Batch Balancing often do not show as significant improvements. ADASYN and SMOTE show moderate improvements but are less effective than class weight. Batch Balanced Over/Under Sampling also shows moderate improvements. Compared to the FT-Transformer, ROS and RUS are not less effective and perform well in electricity dataset. Noise reduction techniques (ENN, RENN, All KNN, and Batch Stratified) continue to show very poor performance with ResNet.

Synthetically Imbalanced Dataset with Lower Imbalance Ratio

As the imbalance ratio decreases (IR 10 and IR 2), the baseline F1 score improves, but not as significantly as in XGBoost. For electricity dataset, the difference in F1 scores achieved by ResNet and FT-Transformer gradually disappears. Similar to the FT-Transformer, class weight remains a strong performer, but ROS and Batch Balancing methods sometimes outperform it, though not significantly. Noise re-

Table 4.6: Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Minority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.035 ± 0.014	0.731 ± 0.008	0.439 ± 0.038	0.482 ± 0.050	0.446 ± 0.010	0.618 ± 0.053	0.036 ± 0.005	0.048 ± 0.003	0.046 ± 0.008	0.464 ± 0.016	0.507 ± 0.037	0.029 ± 0.012
electricity	20	0.302 ± 0.026	0.762 ± 0.006	0.723 ± 0.010	0.739 ± 0.000	0.737 ± 0.006	0.743 ± 0.011	0.320 ± 0.014	0.361 ± 0.038	0.346 ± 0.006	0.742 ± 0.010	0.763 ± 0.008	0.336 ± 0.064
electricity	10	0.487 ± 0.020	0.770 ± 0.006	0.775 ± 0.007	0.777 ± 0.003	0.781 ± 0.007	0.743 ± 0.024	0.556 ± 0.022	0.559 ± 0.019	0.502 ± 0.011	0.785 ± 0.006	0.774 ± 0.006	0.471 ± 0.039
electricity	2	0.782 ± 0.004	0.807 ± 0.001	0.814 ± 0.002	0.800 ± 0.008	0.804 ± 0.007	0.779 ± 0.010	0.783 ± 0.010	0.779 ± 0.003	0.775 ± 0.004	0.809 ± 0.005	0.810 ± 0.007	0.790 ± 0.006
electricity	1	0.814 ± 0.008											
MagicTelescope	100	0.275 ± 0.019	0.738 ± 0.013	0.535 ± 0.030	0.518 ± 0.020	0.493 ± 0.025	0.740 ± 0.017	0.331 ± 0.006	0.343 ± 0.014	0.315 ± 0.021	0.513 ± 0.028	0.517 ± 0.032	0.282 ± 0.034
MagicTelescope	20	0.521 ± 0.005	0.778 ± 0.002	0.729 ± 0.013	0.710 ± 0.019	0.673 ± 0.025	0.766 ± 0.005	0.564 ± 0.007	0.577 ± 0.011	0.587 ± 0.030	0.720 ± 0.008	0.725 ± 0.016	0.510 ± 0.004
MagicTelescope	10	0.638 ± 0.044	0.794 ± 0.002	0.782 ± 0.007	0.779 ± 0.005	0.751 ± 0.012	0.773 ± 0.002	0.663 ± 0.014	0.675 ± 0.014	0.669 ± 0.009	0.800 ± 0.001	0.769 ± 0.003	0.624 ± 0.015
MagicTelescope	2	0.821 ± 0.002	0.831 ± 0.003	0.836 ± 0.001	0.829 ± 0.001	0.825 ± 0.016	0.813 ± 0.012	0.801 ± 0.002	0.794 ± 0.006	0.799 ± 0.001	0.838 ± 0.002	0.840 ± 0.003	0.810 ± 0.004
MagicTelescope	1	0.833 ± 0.006											
GMSC	17	0.031 ± 0.019	0.621 ± 0.056	0.462 ± 0.177	0.383 ± 0.033	0.596 ± 0.039	0.506 ± 0.093	0.055 ± 0.031	0.074 ± 0.060	0.042 ± 0.016	0.530 ± 0.194	0.444 ± 0.102	0.031 ± 0.001
ISCXIDS2012	55	0.000 ± 0.000	0.878 ± 0.060	0.336 ± 0.433	0.043 ± 0.017	0.284 ± 0.397	0.267 ± 0.376	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.642 ± 0.411	0.020 ± 0.028	0.000 ± 0.000

Table 4.7: Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for ResNet. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.696	0.403	0.447	0.410	0.583	0.001	0.013	0.010	0.429	0.472	-0.007
electricity	20	0.460	0.420	0.436	0.435	0.440	0.017	0.058	0.044	0.439	0.460	0.034
electricity	10	0.283	0.288	0.290	0.294	0.256	0.069	0.072	0.015	0.298	0.287	-0.016
electricity	2	0.024	0.032	0.018	0.022	-0.003	0.001	-0.003	-0.007	0.027	0.028	0.008
MagicTelescope	100	0.463	0.260	0.243	0.218	0.465	0.056	0.068	0.040	0.238	0.242	0.008
MagicTelescope	20	0.257	0.208	0.189	0.152	0.245	0.043	0.056	0.066	0.199	0.204	-0.012
MagicTelescope	10	0.156	0.143	0.141	0.113	0.135	0.025	0.037	0.031	0.162	0.131	-0.014
MagicTelescope	2	0.009	0.015	0.008	0.004	-0.009	-0.021	-0.027	-0.023	0.017	0.019	-0.011
GMSC	17	0.590	0.432	0.352	0.565	0.475	0.025	0.043	0.012	0.500	0.413	0.001
ISCXIDS2012	55	0.878	0.336	0.043	0.284	0.267	0.000	0.000	0.000	0.642	0.020	0.000

duction techniques continue to underperform, and with the lowest imbalance ratio of 2, the F1 score for majority class generally decreases.

Real-World Dataset

In the GiveMeSomeCredit dataset, the baseline performance is low, and lower than the score achieved by the FT-Transformer. Class weight achieves the best improvement in the F1 score, but the F1 score is still not comparable to that of the FT-Transformer. This pattern holds true across other methods as well; although different class imbalance methods mitigate the impact of class imbalance on model training, the improvements do not surpass those achieved by the FT-Transformer using the same methods. In the ISCXIDS2012 dataset, the baseline F1 score is 0, the same as with the FT-Transformer. Class weight shows the most significant improvements among different class imbalance handling methods. Compared to the best performance of the FT-Transformer, ResNet with class weight outperforms it. However, other class imbalance handling methods do not effectively address the impact of class imbalance in this dataset on model training for ResNet, marking a notable difference from the FT-Transformer.

The best methods across various imbalance ratios and datasets for ResNet are consistently class weight. Batch Balancing methods, ROS and RUS provide moderate improvements, while noise reduction techniques (ENN, RENN, All KNN) and Batch Stratified methods often underperform. These performance trends are consistent across controlled datasets (electricity and MagicTelescope) and real-world datasets (GiveMeSomeCredit and ISCXIDS2012).

Regarding both the F1 score for the minority class and the majority class, RUS, noise reduction techniques (ENN, RENN, All KNN) and batch stratified consistently show a pattern where a decrease in the F1 score for the minority class is

Table 4.8: Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Majority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.671 ± 0.002	0.726 ± 0.050	0.720 ± 0.007	0.728 ± 0.012	0.722 ± 0.004	0.618 ± 0.021	0.671 ± 0.001	0.672 ± 0.000	0.672 ± 0.001	0.722 ± 0.006	0.728 ± 0.004	0.670 ± 0.001
electricity	20	0.707 ± 0.004	0.770 ± 0.008	0.777 ± 0.005	0.782 ± 0.001	0.779 ± 0.001	0.763 ± 0.004	0.710 ± 0.002	0.717 ± 0.007	0.713 ± 0.001	0.780 ± 0.004	0.787 ± 0.004	0.713 ± 0.011
electricity	10	0.742 ± 0.005	0.782 ± 0.004	0.790 ± 0.008	0.793 ± 0.003	0.792 ± 0.005	0.767 ± 0.006	0.756 ± 0.005	0.756 ± 0.004	0.743 ± 0.002	0.794 ± 0.004	0.791 ± 0.003	0.739 ± 0.008
electricity	2	0.821 ± 0.002	0.812 ± 0.005	0.819 ± 0.005	0.807 ± 0.002	0.803 ± 0.008	0.793 ± 0.002	0.786 ± 0.003	0.751 ± 0.011	0.770 ± 0.007	0.815 ± 0.004	0.808 ± 0.008	0.821 ± 0.001
electricity	1	0.807 ± 0.012											
MagicTelescope	100	0.704 ± 0.003	0.799 ± 0.003	0.750 ± 0.005	0.743 ± 0.004	0.736 ± 0.005	0.771 ± 0.010	0.712 ± 0.001	0.715 ± 0.002	0.710 ± 0.004	0.747 ± 0.007	0.747 ± 0.008	0.705 ± 0.006
MagicTelescope	20	0.754 ± 0.002	0.808 ± 0.006	0.806 ± 0.007	0.790 ± 0.009	0.782 ± 0.006	0.797 ± 0.001	0.764 ± 0.002	0.768 ± 0.003	0.771 ± 0.008	0.802 ± 0.005	0.798 ± 0.004	0.751 ± 0.001
MagicTelescope	10	0.788 ± 0.015	0.808 ± 0.007	0.821 ± 0.007	0.815 ± 0.006	0.796 ± 0.008	0.797 ± 0.002	0.793 ± 0.005	0.795 ± 0.004	0.795 ± 0.002	0.826 ± 0.009	0.809 ± 0.002	0.782 ± 0.005
MagicTelescope	2	0.855 ± 0.002	0.854 ± 0.002	0.856 ± 0.003	0.852 ± 0.001	0.827 ± 0.033	0.833 ± 0.014	0.800 ± 0.007	0.761 ± 0.006	0.783 ± 0.003	0.857 ± 0.000	0.853 ± 0.001	0.851 ± 0.002
MagicTelescope	1	0.852 ± 0.004											
GMSC	17	0.669 ± 0.002	0.731 ± 0.018	0.652 ± 0.074	0.701 ± 0.004	0.651 ± 0.051	0.565 ± 0.064	0.672 ± 0.003	0.674 ± 0.007	0.671 ± 0.002	0.698 ± 0.041	0.701 ± 0.029	0.670 ± 0.000
ISCXIDS2012	55	0.667 ± 0.000	0.823 ± 0.122	0.760 ± 0.129	0.667 ± 0.000	0.730 ± 0.090	0.671 ± 0.016	0.667 ± 0.000	0.667 ± 0.000	0.667 ± 0.000	0.839 ± 0.124	0.665 ± 0.003	0.667 ± 0.000

Table 4.9: Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for ResNet. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.056	0.049	0.057	0.052	-0.053	0.000	0.001	0.001	0.052	0.057	-0.001
electricity	20	0.063	0.070	0.075	0.072	0.056	0.003	0.010	0.006	0.073	0.080	0.006
electricity	10	0.041	0.049	0.051	0.051	0.025	0.015	0.014	0.001	0.053	0.049	-0.003
electricity	2	-0.009	-0.002	-0.014	-0.019	-0.029	-0.036	-0.071	-0.052	-0.006	-0.013	-0.000
MagicTelescope	100	0.096	0.047	0.040	0.032	0.068	0.009	0.011	0.006	0.043	0.043	0.001
MagicTelescope	20	0.054	0.052	0.037	0.029	0.044	0.011	0.014	0.017	0.048	0.044	-0.003
MagicTelescope	10	0.020	0.034	0.028	0.009	0.009	0.005	0.007	0.007	0.038	0.021	-0.005
MagicTelescope	2	-0.000	0.001	-0.003	-0.028	-0.021	-0.055	-0.094	-0.072	0.002	-0.002	-0.003
GMSC	17	0.062	-0.018	0.032	-0.018	-0.105	0.003	0.005	0.001	0.029	0.032	0.000
ISCXIDS2012	55	0.157	0.093	0.001	0.063	0.005	0.000	0.000	0.000	0.172	-0.002	0.000

accompanied by a decrease in the F1 score for the majority class.

4.1.3 Detailed Analysis of XGBoost

Finally, for XGBoost, the F1 scores for the minority class, majority class, and macro average are presented in Table 4.11, Table 4.13, and Table 4.15. Similar to the DL models, the improvements for both F1 scores of the minority class and the majority class for each method are presented in Table 4.12 and Table 4.14.

Synthetically Imbalanced Dataset with Higher Imbalance Ratio

For datasets with a high imbalance ratio (IR 100 and IR 2), even with XGBoost's inherent capabilities to handle class imbalance, the baseline F1 score is still low. The RUS method consistently performs best among the various methods. The class weight method, while effective, does not bring as significant improvements as seen with DL models like ResNet and FT-Transformer. ADASYN and SMOTE show substantial improvements over the baseline but are less effective than RUS, with SMOTE and ADASYN outperforming class weight in datasets with the highest imbalance ratio (IR 100). Noise reduction techniques (ENN, RENN, and All KNN) show less significant performance improvements compared to other methods.

Synthetically Imbalanced Dataset with Lower Imbalance Ratio

For datasets with lower imbalance ratios (IR 10 and IR 2), XGBoost demonstrates its ability to handle class imbalance efficiently without any specific class imbalance handling methods, as indicated by the high F1 scores for no treatment. The difference in the efficacy of different class imbalance handling methods becomes less significant for datasets with lower imbalance ratio. While RUS remains the top performer for higher imbalance ratio datasets, its advantage diminishes as the

Table 4.10: Empirical Comparison of Class Imbalance Handling Method for ResNet (F1 Score Macro). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	0.353 ± 0.008	0.729 ± 0.021	0.579 ± 0.023	0.605 ± 0.031	0.584 ± 0.007	0.618 ± 0.036	0.354 ± 0.003	0.360 ± 0.001	0.359 ± 0.004	0.593 ± 0.011	0.617 ± 0.019	0.349 ± 0.007
electricity	20	0.505 ± 0.015	0.766 ± 0.005	0.750 ± 0.007	0.760 ± 0.001	0.758 ± 0.003	0.753 ± 0.007	0.515 ± 0.008	0.539 ± 0.022	0.530 ± 0.003	0.761 ± 0.003	0.775 ± 0.006	0.524 ± 0.038
electricity	10	0.614 ± 0.013	0.776 ± 0.004	0.782 ± 0.008	0.785 ± 0.003	0.787 ± 0.001	0.755 ± 0.014	0.656 ± 0.014	0.658 ± 0.011	0.622 ± 0.006	0.790 ± 0.005	0.782 ± 0.003	0.605 ± 0.024
electricity	2	0.802 ± 0.003	0.809 ± 0.002	0.817 ± 0.003	0.803 ± 0.005	0.803 ± 0.007	0.786 ± 0.005	0.784 ± 0.005	0.765 ± 0.005	0.772 ± 0.004	0.812 ± 0.005	0.809 ± 0.007	0.806 ± 0.003
electricity	1	0.810 ± 0.010											
MagicTelescope	100	0.489 ± 0.011	0.769 ± 0.008	0.643 ± 0.018	0.631 ± 0.012	0.615 ± 0.015	0.756 ± 0.012	0.522 ± 0.004	0.529 ± 0.008	0.512 ± 0.013	0.630 ± 0.017	0.632 ± 0.020	0.494 ± 0.020
MagicTelescope	20	0.637 ± 0.003	0.793 ± 0.003	0.768 ± 0.010	0.750 ± 0.012	0.728 ± 0.015	0.782 ± 0.003	0.664 ± 0.005	0.673 ± 0.007	0.679 ± 0.019	0.761 ± 0.007	0.761 ± 0.010	0.630 ± 0.002
MagicTelescope	10	0.713 ± 0.030	0.801 ± 0.004	0.801 ± 0.007	0.797 ± 0.005	0.774 ± 0.008	0.785 ± 0.002	0.728 ± 0.009	0.735 ± 0.009	0.732 ± 0.006	0.813 ± 0.005	0.789 ± 0.002	0.703 ± 0.010
MagicTelescope	2	0.838 ± 0.002	0.843 ± 0.002	0.846 ± 0.001	0.841 ± 0.001	0.826 ± 0.024	0.823 ± 0.013	0.800 ± 0.003	0.778 ± 0.006	0.791 ± 0.002	0.847 ± 0.001	0.846 ± 0.001	0.831 ± 0.002
MagicTelescope	1	0.842 ± 0.004											
GMSC	17	0.350 ± 0.011	0.676 ± 0.022	0.557 ± 0.052	0.542 ± 0.018	0.623 ± 0.007	0.535 ± 0.016	0.364 ± 0.017	0.374 ± 0.033	0.356 ± 0.009	0.614 ± 0.094	0.572 ± 0.057	0.350 ± 0.000
ISCXIDS2012	55	0.333 ± 0.000	0.851 ± 0.091	0.548 ± 0.281	0.355 ± 0.009	0.507 ± 0.243	0.469 ± 0.196	0.333 ± 0.000	0.333 ± 0.000	0.333 ± 0.000	0.740 ± 0.267	0.342 ± 0.013	0.333 ± 0.000

Table 4.11: Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Minority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	0.103 ± 0.038	0.285 ± 0.006	0.278 ± 0.017	0.415 ± 0.031	0.438 ± 0.026	0.762 ± 0.013	0.192 ± 0.004	0.193 ± 0.006	0.137 ± 0.022
electricity	20	0.561 ± 0.006	0.648 ± 0.007	0.649 ± 0.007	0.676 ± 0.006	0.679 ± 0.005	0.807 ± 0.002	0.597 ± 0.013	0.607 ± 0.004	0.601 ± 0.010
electricity	10	0.703 ± 0.003	0.768 ± 0.005	0.761 ± 0.004	0.774 ± 0.004	0.778 ± 0.003	0.839 ± 0.004	0.745 ± 0.001	0.753 ± 0.001	0.753 ± 0.001
electricity	2	0.900 ± 0.003	0.907 ± 0.001	0.906 ± 0.003	0.904 ± 0.004	0.902 ± 0.002	0.904 ± 0.004	0.889 ± 0.002	0.864 ± 0.002	0.876 ± 0.002
electricity	1	0.915 ± 0.000								
MagicTelescope	100	0.316 ± 0.026	0.459 ± 0.034	0.450 ± 0.012	0.624 ± 0.021	0.626 ± 0.017	0.763 ± 0.012	0.395 ± 0.017	0.408 ± 0.019	0.390 ± 0.015
MagicTelescope	20	0.608 ± 0.002	0.756 ± 0.002	0.665 ± 0.004	0.754 ± 0.005	0.757 ± 0.003	0.801 ± 0.004	0.671 ± 0.002	0.681 ± 0.004	0.675 ± 0.004
MagicTelescope	10	0.692 ± 0.003	0.738 ± 0.005	0.742 ± 0.005	0.790 ± 0.005	0.790 ± 0.001	0.817 ± 0.003	0.750 ± 0.006	0.765 ± 0.005	0.762 ± 0.003
MagicTelescope	2	0.834 ± 0.002	0.841 ± 0.002	0.838 ± 0.001	0.845 ± 0.000	0.843 ± 0.002	0.847 ± 0.006	0.840 ± 0.004	0.820 ± 0.004	0.835 ± 0.003
MagicTelescope	1	0.843 ± 0.000								
GMSC	17	0.266 ± 0.010	0.467 ± 0.001	0.392 ± 0.004	0.434 ± 0.011	0.436 ± 0.003	0.779 ± 0.002	0.303 ± 0.003	0.334 ± 0.002	0.317 ± 0.007
ISCXIDS2012	55	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.001	0.999 ± 0.000	0.998 ± 0.000	0.998 ± 0.000	0.998 ± 0.001

Table 4.12: Difference to No Treatment in F1 Scores for Minority Classes After Applying Class Imbalance Handling Methods for XGBoost. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	0.182	0.175	0.312	0.335	0.659	0.088	0.090	0.034
electricity	20	0.087	0.088	0.115	0.118	0.246	0.036	0.046	0.040
electricity	10	0.065	0.058	0.072	0.075	0.137	0.043	0.051	0.051
electricity	2	0.007	0.007	0.004	0.002	0.004	-0.011	-0.036	-0.023
MagicTelescope	100	0.143	0.134	0.308	0.310	0.447	0.079	0.092	0.074
MagicTelescope	20	0.147	0.057	0.146	0.148	0.193	0.063	0.072	0.066
MagicTelescope	10	0.046	0.050	0.098	0.098	0.126	0.058	0.073	0.070
MagicTelescope	2	0.007	0.004	0.011	0.009	0.013	0.006	-0.014	0.001
GMSC	17	0.200	0.125	0.168	0.169	0.513	0.037	0.068	0.050
ISCXIDS2012	55	0.000	0.000	0.000	-0.000	0.001	-0.000	-0.000	-0.000

imbalance ratio decreases. Class weight shows a strong performance but is slightly outperformed by methods like ADASYN and SMOTE in certain scenarios. Noise reduction techniques continue to underperform across these datasets.

Real-World Dataset

In the GiveMeSomeCredit dataset, RUS shows the best performance in handling class imbalance, followed by class weight. The improvement with class weight is substantial but not as significant as with RUS. In the ISCXIDS2012 dataset, despite a higher imbalance ratio than GiveMeSomeCredit, the baseline F1 score is already high, resulting in generally high F1 scores with no significant improvements regardless of the class imbalance method used.

Unlike DL methods, the RUS method generally performs best with XGBoost, except for the electricity dataset with IR 2. Even for electricity with IR 2, the performance improvement by RUS is comparable to the best performance achieved by class weight. Noise reduction techniques show less improvement with XGBoost compared to oversampling and class weight methods. In fact, these techniques can detrimentally impact the F1 score for the minority class, as seen in the electricity dataset with IR 2 and ISCXIDS2012, along with decreasing the F1 score for the majority class.

Table 4.13: Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Majority Class). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	0.679 ± 0.005	0.705 ± 0.001	0.704 ± 0.003	0.729 ± 0.007	0.734 ± 0.006	0.769 ± 0.012	0.691 ± 0.001	0.691 ± 0.001	0.683 ± 0.003
electricity	20	0.765 ± 0.002	0.788 ± 0.003	0.790 ± 0.003	0.798 ± 0.002	0.800 ± 0.003	0.809 ± 0.001	0.774 ± 0.004	0.776 ± 0.001	0.775 ± 0.003
electricity	10	0.810 ± 0.002	0.833 ± 0.003	0.833 ± 0.003	0.838 ± 0.002	0.839 ± 0.001	0.838 ± 0.005	0.823 ± 0.001	0.825 ± 0.001	0.825 ± 0.001
electricity	2	0.909 ± 0.002	0.913 ± 0.002	0.912 ± 0.002	0.909 ± 0.003	0.908 ± 0.001	0.903 ± 0.004	0.881 ± 0.002	0.837 ± 0.002	0.860 ± 0.002
electricity	1	0.915 ± 0.000								
MagicTelescope	100	0.711 ± 0.005	0.739 ± 0.008	0.737 ± 0.003	0.779 ± 0.007	0.780 ± 0.006	0.778 ± 0.013	0.725 ± 0.004	0.728 ± 0.004	0.724 ± 0.003
MagicTelescope	20	0.778 ± 0.001	0.821 ± 0.002	0.796 ± 0.002	0.825 ± 0.002	0.825 ± 0.001	0.808 ± 0.010	0.798 ± 0.001	0.801 ± 0.002	0.799 ± 0.002
MagicTelescope	10	0.806 ± 0.001	0.821 ± 0.003	0.824 ± 0.002	0.839 ± 0.003	0.837 ± 0.001	0.826 ± 0.004	0.827 ± 0.003	0.831 ± 0.002	0.831 ± 0.001
MagicTelescope	2	0.862 ± 0.001	0.862 ± 0.002	0.861 ± 0.001	0.860 ± 0.001	0.856 ± 0.002	0.853 ± 0.006	0.839 ± 0.006	0.786 ± 0.007	0.820 ± 0.005
MagicTelescope	1	0.849 ± 0.001								
GMSC	17	0.699 ± 0.001	0.718 ± 0.003	0.708 ± 0.002	0.708 ± 0.003	0.709 ± 0.002	0.785 ± 0.001	0.704 ± 0.001	0.709 ± 0.001	0.706 ± 0.001
ISCXIDS2012	55	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.001	0.999 ± 0.000	0.998 ± 0.000	0.998 ± 0.000	0.998 ± 0.001

Table 4.14: Difference to No Treatment in F1 Scores for Majority Classes After Applying Class Imbalance Handling Methods for XGBoost. Negative improvements are highlighted in Red.

Dataset	IR	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	0.026	0.025	0.050	0.055	0.090	0.012	0.012	0.004
electricity	20	0.024	0.026	0.034	0.035	0.044	0.009	0.012	0.010
electricity	10	0.023	0.023	0.028	0.029	0.028	0.013	0.015	0.015
electricity	2	0.004	0.004	0.000	-0.001	-0.006	-0.028	-0.072	-0.048
MagicTelescope	100	0.028	0.026	0.068	0.068	0.067	0.014	0.017	0.013
MagicTelescope	20	0.043	0.017	0.046	0.047	0.030	0.020	0.023	0.021
MagicTelescope	10	0.015	0.018	0.033	0.031	0.020	0.021	0.025	0.024
MagicTelescope	2	0.000	-0.001	-0.002	-0.006	-0.008	-0.023	-0.076	-0.042
GMSC	17	0.019	0.009	0.009	0.010	0.087	0.005	0.010	0.007
ISCXIDS2012	55	0.000	0.000	0.000	0.000	0.001	-0.000	-0.000	-0.000

Table 4.15: Empirical Comparison of Class Imbalance Handling Method for XGBoost (F1 Score Macro). For each dataset, top result is highlighted in Green.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	0.391 ± 0.005	0.495 ± 0.001	0.491 ± 0.003	0.572 ± 0.007	0.586 ± 0.006	0.765 ± 0.012	0.441 ± 0.001	0.442 ± 0.001	0.410 ± 0.003
electricity	20	0.663 ± 0.002	0.718 ± 0.003	0.720 ± 0.003	0.737 ± 0.002	0.739 ± 0.003	0.808 ± 0.001	0.685 ± 0.004	0.692 ± 0.001	0.688 ± 0.003
electricity	10	0.756 ± 0.002	0.800 ± 0.003	0.797 ± 0.003	0.806 ± 0.002	0.808 ± 0.001	0.839 ± 0.005	0.784 ± 0.001	0.789 ± 0.001	0.789 ± 0.001
electricity	2	0.904 ± 0.002	0.910 ± 0.002	0.909 ± 0.002	0.906 ± 0.003	0.905 ± 0.001	0.903 ± 0.004	0.885 ± 0.002	0.850 ± 0.002	0.868 ± 0.002
electricity	1	0.915 ± 0.000								
MagicTelescope	100	0.514 ± 0.005	0.599 ± 0.008	0.593 ± 0.003	0.702 ± 0.007	0.703 ± 0.006	0.771 ± 0.013	0.560 ± 0.004	0.568 ± 0.004	0.557 ± 0.003
MagicTelescope	20	0.693 ± 0.001	0.789 ± 0.002	0.730 ± 0.002	0.789 ± 0.002	0.791 ± 0.001	0.805 ± 0.010	0.735 ± 0.001	0.741 ± 0.002	0.737 ± 0.002
MagicTelescope	10	0.749 ± 0.001	0.779 ± 0.003	0.783 ± 0.002	0.815 ± 0.003	0.814 ± 0.001	0.822 ± 0.004	0.788 ± 0.003	0.798 ± 0.002	0.796 ± 0.001
MagicTelescope	2	0.848 ± 0.001	0.851 ± 0.002	0.849 ± 0.001	0.852 ± 0.001	0.849 ± 0.002	0.850 ± 0.006	0.840 ± 0.006	0.803 ± 0.007	0.827 ± 0.005
MagicTelescope	1	0.846 ± 0.001								
GMSC	17	0.483 ± 0.001	0.592 ± 0.003	0.550 ± 0.002	0.571 ± 0.003	0.573 ± 0.002	0.782 ± 0.001	0.504 ± 0.001	0.521 ± 0.001	0.511 ± 0.001
ISCXIDS2012	55	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.999 ± 0.000	0.998 ± 0.001	0.999 ± 0.000	0.998 ± 0.000	0.998 ± 0.000	0.998 ± 0.001

Chapter 5

Discussion

5.1 Impact of Different Class Imbalance Handling Methods on Deep Learning Models

In this section, I explore the impact of various class imbalance handling techniques on the performance of the FT-Transformer, ResNet, and XGBoost. The goal is to compare the effectiveness of these techniques in mitigating the impact of class imbalance within the same model. Additionally, I analyze whether the comparative effectiveness of these techniques differs across different models and, if so, how it differs. The rankings of these methods for each dataset and model are summarized in Table 5.1, Table 5.2, and Table 5.3.

Over-sampling vs. Under-sampling In the conducted experiments, over-sampling methods such as ROS, SMOTE, and ADASYN generally outperformed under-sampling methods, specifically noise reduction techniques like ENN, RENN, and All KNN. This trend was consistent across multiple datasets and imbalance ratios, with over-sampling techniques yielding higher F1 scores. The effectiveness of over-sampling can be attributed to the increased representation of minority class samples without the loss of majority class data, which is often a limitation of under-sampling techniques.

ROS vs. RUS The experiments revealed no significant evidence favoring ROS or RUS over the other in handling class imbalance across the models tested, specifically the FT-Transformer. When trained on 10 imbalanced datasets, each method outperformed the other on 5 datasets (a half of datasets) for this model. This parity suggests that both methods can be effective, but their performance may vary depending on the specific characteristics of the dataset for the FT-Transformer.

Table 5.1: Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in FT-Transformer. For each dataset, top 3 rank methods are highlighted in Green.

dataset	electricity				MagicTelescope				GMSC	ISCXIDS2012	Avg Rank	Total Rank
IR	100	20	10	2	100	20	10	2	17	55		
None	11	12	11	9	11	12	10	8	9	10	10.3	11
CW	1	1	2	2	1	1	4	7	1	4	2.4	1
ROS	7	4	4	6	6	6	6	3	2	6	5	7
SMOTE	3	3	3	1	3	3	5	5	5	5	3.6	3
ADASYN	2	5	1	3	4	5	2	4	7	7	4	4
RUS	6	6	7	7	2	2	7	2	6	3	4.8	6
ENN	10	8	12	12	9	8	9	9	11	10	9.8	9
RENN	12	9	9	11	10	11	12	12	8	10	10.4	12
ALL KNN	9	11	8	10	8	9	8	11	12	10	9.6	8
BB OS	5	2	5	4	5	7	1	1	4	1	3.5	2
BB US	4	7	6	5	7	4	3	6	3	2	4.7	5
BS	8	10	10	8	12	10	11	10	10	10	9.9	10

Table 5.2: Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in ResNet. For each dataset, top 3 rank methods are highlighted in Green.

dataset	electricity				MagicTelescope				GMSC	ISCXIDS2012	Avg Rank	Total Rank
IR	100	20	10	2	100	20	10	2	17	55		
None	11	12	11	9	12	11	11	7	12	10	9	9
CW	1	2	6	4	2	1	2	4	1	1	2.4	1
ROS	7	7	4	1	3	3	3	3	5	3	3.9	3
SMOTE	4	5	3	6	4	6	4	5	7	6	5	6
ADASYN	6	6	2	5	7	7	7	6	2	4	5.2	7
RUS	2	3	7	11	1	2	5	8	4	5	4.8	5
ENN	10	11	9	8	9	10	10	10	9	10	9.6	10
RENN	8	8	8	10	8	9	8	12	8	10	8.9	8
ALL KNN	9	9	10	12	10	8	9	11	10	10	9.8	11
BB OS	5	4	1	3	6	5	1	2	3	2	3.2	2
BB US	3	1	5	2	5	4	6	1	6	7	4	4
BS	12	10	12	7	11	12	12	9	11	10	10.6	12

Table 5.3: Rank of F1 Score for Minority Class Using Various Class Imbalance Handling Techniques in XGBoost. For each dataset, top 3 rank methods are highlighted in Green.

dataset	electricity				MagicTelescope				GMSC	ISCXIDS2012	Avg Rank	Total Rank
IR	100	20	10	2	100	20	10	2	17	55		
NONE	9	9	9	6	9	9	9	8	9	4	8.1	9
CW	4	5	4	1	4	3	8	4	2	2	3.7	4
ROS	5	4	5	2	5	8	7	6	5	4	5.1	5
SMOTE	3	3	3	4	3	4	3	2	4	2	3.1	3
ADASYN	2	2	2	5	2	2	2	3	3	6	2.9	2
RUS	1	1	1	3	1	1	1	1	1	1	1.2	1
ENN	7	8	8	7	7	7	6	5	8	8	7.1	7
RENN	6	6	7	9	6	5	4	9	6	7	6.5	6
ALL KNN	8	7	6	8	8	6	5	7	7	9	7.1	7

In contrast, for ResNet, a pattern emerged in synthetically controlled imbalanced datasets. With a higher imbalance ratio (IR 100 and IR 20), RUS performs better than ROS. Conversely, with a lower imbalance ratio (IR 10 and IR 2), ROS outperforms RUS. This suggests that the imbalance ratio of the dataset might influence which method, ROS or RUS, works better for ResNet.

ROS vs. SMOTE vs. ADASYN For the FT-Transformer, SMOTE outperformed ROS in 8 out of 10 datasets, whereas for ResNet, SMOTE only outperformed ROS in 3 out of 10 datasets. Similarly, ADASYN outperformed ROS in 7 cases for the FT-Transformer but was outperformed by ROS in 6 cases for ResNet. This indicates that while SMOTE and ADASYN may have advantages in certain scenarios, their effectiveness is not consistent across different models. Additionally, ADASYN and SMOTE exhibit a similar tendency in terms of outperforming or underperforming ROS. However, the differences in performance between SMOTE and ADASYN do not show a clear pattern, making it difficult to determine a universally superior method.

Balanced Number of Samples Both ROS and RUS demonstrated notable improvements in handling class imbalance, emphasizing the importance of balanced sample sizes. This is reflected in their higher F1 scores, indicating that maintaining an equilibrium in sample sizes for both classes is critical for model performance. The balanced sample sizes prevent the model from being biased towards the majority class, thereby enhancing the detection of minority class instances.

Noise Reduction Techniques Noise reduction techniques such as ENN, RENN, and All KNN generally showed poor performance compared to other methods.

They often resulted in lower F1 scores, and in some cases, worsened performance relative to datasets with no class imbalance treatment. This can be attributed to the removal of data points, which may include critical information necessary for accurate classification, thus reducing overall data representation and potentially increasing bias.

Although RUS also removes data samples from the majority class, noise reduction methods have specific criteria for removing samples. These methods do not aim to balance the number of classes but rather to eliminate noise that may interfere with building accurate decision boundaries. However, this does not result in an equal number of samples across classes. From this perspective and based on experimental results, noise points in imbalanced datasets do not significantly detract from model performance. Consequently, methods that remove samples based on their surrounding neighbors are not the most effective solutions for addressing class imbalance issues. While there is research [22] indicating that class imbalance can involve additional challenges such as noise, overlapping data, and small disjuncts, the discrepancy in the number of samples remains the most impactful factor in deteriorating model performance. Therefore, focusing on balancing the sample sizes between classes is more critical for improving model performance than solely addressing noise within the dataset.

Class Weight Techniques Class weight adjustments consistently improved model performance, providing significant F1 score enhancements across most scenarios. These techniques assign higher weights to minority class samples during training, ensuring that the model pays more attention to these instances. This approach was effective across both FT-Transformer and ResNet models, as well as in different datasets, including controlled and real-world datasets. This demonstrates its broad applicability and robustness in handling class imbalance.

Batch Balancing Methods Batch balancing, which involves adjusting mini-batch compositions to ensure a balanced representation of classes, also achieved good results. Although class weight adjustments sometimes outperformed batch balancing in highly imbalanced scenarios, batch balancing methods still provided substantial improvements. There was no clear preference between over-sampling and under-sampling within batch balancing, as both approaches yielded similar enhancements in F1 scores for minority classes.

Based on the observation, applying appropriate approaches that address class imbalance can significantly enhance the performance and reliability of DL models in class-imbalanced scenarios. Although the optimal solution can vary depending

on the specific model and dataset, more effective methods under certain conditions were recognized. Overall, the empirical comparison emphasizes that balancing the importance of each class is the most effective method to mitigate the impact of class imbalance on DL model training. This is supported by the results using class weights in the loss function during the training process and other methods that balance minority and majority samples before training. The trends observed in controlled datasets (electricity and MagicTelescope) and in real-world datasets (GiveMeSomeCredit and ISCXIDS2012) are aligned, supporting the broader applicability of the findings.

5.2 Comparative Analysis: FT-Transformer vs. ResNet vs. XGBoost

This section presents a comparative analysis of the performance of various class imbalance handling techniques on three models: FT-Transformer, ResNet, and XGBoost. The model comparison is done in two ways. First, I compare the best F1 score for each model across different datasets. Second, I evaluate how different the effectiveness of each method is on different models by subtracting the F1 scores of each method for one model from those of another model. This dual comparison allows me to determine which model performs better after applying class imbalance handling techniques and to identify which class imbalance methods are most effective for each model. The first comparison, showing the best F1 score for each model, can be found in Table 5.4. The second comparison results are shown in Table 5.5 (FT-Transformer vs. ResNet), Table 5.6 (FT-Transformer vs. XGBoost), and the Table 5.7 (ResNet vs. XGBoost).

In Table 5.5, the results show that overall, ResNet generally performs better than the FT-Transformer in terms of F1 scores for the minority class. However, in real-world datasets like GiveMeSomeCredit and ISCXIDS2012, the FT-Transformer shows superior performance. This indicates that the FT-Transformer may have advantages in handling more complex, real-world data scenarios.

In Table 5.6, the comparison between the FT-Transformer and XGBoost reveals that while XGBoost often outperforms the FT-Transformer in baseline scenarios, certain class imbalance handling methods like class weight and ROS can allow the FT-Transformer to achieve comparable or even superior performance in some datasets with higher imbalance ratios.

Table 5.7 shows similar trends between ResNet and XGBoost, with XGBoost generally performing better. However, when applying class imbalance handling

Dataset	IR	FT-Transformer	ResNet	XGBoost	XGBoost (no treatment)
electricity	100	0.730	0.731	0.762	0.103
electricity	20	0.752	0.763	0.807	0.561
electricity	10	0.777	0.785	0.839	0.703
electricity	2	0.825	0.814	0.907	0.900
electricity	1	0.805	0.814	0.915	0.915
MagicTelescope	100	0.741	0.740	0.763	0.316
MagicTelescope	20	0.772	0.778	0.801	0.608
MagicTelescope	10	0.785	0.800	0.817	0.692
MagicTelescope	2	0.829	0.840	0.847	0.834
MagicTelescope	1	0.828	0.833	0.843	0.843
GMSC	17	0.770	0.621	0.779	0.266
ISCXIDS2012	55	0.886	0.878	0.999	0.998

Table 5.4: Comparison the Best F1 Score of Models by Class Imbalance Handling Methods and XGBoost's Baseline F1 Score without Any Class Imbalance Handling Methods Across Different Datasets and Imbalance Ratios (IR). For each dataset, top results between DL models (FT-Transformer and ResNet) are in Green

methods like class weight and ROS, ResNet can sometimes outperform XGBoost in certain datasets.

Additionally, for datasets with the lowest imbalance ratios (e.g., electricity with IR 2 and MagicTelescope with IR 2), XGBoost effectively handles class imbalance internally without additional class imbalance handling methods. XGBoost performs comparably to scenarios where it is trained on balanced datasets. Even with the best performing handling methods for DL models, it is challenging to outperform XGBoost. Based on the comparison between DL models and XGBoost, it can be inferred that certain methods handle class imbalance in datasets more effectively for particular models between DL and ML. Specifically, class weight and ROS are more effective for DL methods in addressing class imbalance. Nonetheless, the best F1 score achieved by XGBoost with class imbalance handling methods still cannot be outperformed by DL models with class imbalance handling methods in any datasets. In terms of DL models, even with the longer training time and more resources required for training, the FT-Transformer underperforms compared to ResNet in most scenarios.

5.2.1 Comparative Analysis: Key Insights

1. FT-Transformer vs. ResNet

    - ResNet generally has a stronger baseline performance compared to the

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN	BB OS	BB US	BS
electricity	100	-0.027	-0.001	-0.108	-0.064	0.015	-0.283	-0.011	-0.044	-0.019	-0.089	-0.129	0.013
electricity	20	-0.097	-0.010	0.004	-0.001	-0.011	-0.023	0.057	-0.021	-0.103	0.005	-0.056	-0.076
electricity	10	-0.002	0.000	-0.014	-0.009	-0.005	-0.009	-0.077	-0.044	0.040	-0.026	-0.018	0.024
electricity	2	0.007	0.018	0.001	0.025	0.015	0.019	-0.002	0.002	0.010	0.007	0.006	0.002
electricity	1	-0.008											
MagicTelescope	100	-0.013	0.003	-0.053	0.031	0.040	-0.072	-0.021	-0.074	-0.002	-0.010	-0.040	-0.039
MagicTelescope	20	0.021	-0.006	-0.024	0.034	0.039	-0.009	-0.002	-0.033	-0.039	-0.017	-0.013	0.037
MagicTelescope	10	-0.005	-0.024	-0.034	-0.023	0.027	-0.032	-0.020	-0.048	-0.020	-0.015	0.006	0.009
MagicTelescope	2	-0.002	-0.009	-0.008	-0.002	0.002	0.016	-0.001	-0.006	-0.010	-0.009	-0.016	-0.013
MagicTelescope	1	-0.004											
GMSC	17	0.212	0.150	0.302	0.356	0.136	0.230	0.184	0.212	0.194	0.228	0.319	0.211
ISCXIDS2012	55		-0.116	0.268	0.564	-0.071	0.499				0.244	0.829	

Table 5.5: Subtraction of FT-Transformer - ResNet in F1 Score for Minority Class, Positive values highlighted in Green

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	-0.094	0.445	0.053	0.003	0.023	-0.427	-0.166	-0.189	-0.111
electricity	20	-0.356	0.104	0.077	0.062	0.047	-0.087	-0.220	-0.268	-0.358
electricity	10	-0.217	0.002	0.000	-0.006	-0.001	-0.105	-0.267	-0.238	-0.211
electricity	2	-0.111	-0.083	-0.091	-0.079	-0.083	-0.106	-0.108	-0.082	-0.092
electricity	1	-0.110								
MagicTelescope	100	-0.054	0.282	0.031	-0.076	-0.094	-0.094	-0.086	-0.140	-0.077
MagicTelescope	20	-0.067	0.017	0.040	-0.010	-0.045	-0.044	-0.109	-0.136	-0.127
MagicTelescope	10	-0.059	0.032	0.006	-0.034	-0.012	-0.076	-0.107	-0.137	-0.112
MagicTelescope	2	-0.014	-0.019	-0.010	-0.017	-0.015	-0.018	-0.040	-0.031	-0.046
MagicTelescope	1	-0.015								
GMSC	17	-0.024	0.304	0.372	0.305	0.296	-0.043	-0.064	-0.049	-0.081
ISCXIDS2012	55	-0.998	-0.236	-0.394	-0.392	-0.785	-0.233	-0.998	-0.998	-0.998

Table 5.6: Subtraction of FT-Transformer - XGBoost in F1 Score for Minority Class, Positive values highlighted in Green

FT-Transformer.

- The FT-Transformer shows better performance in real-world datasets.
- Class imbalance handling methods like class weight and ROS significantly enhance the performance of DL models.

2. DL models (FT-Transformer & ResNet) vs. XGBoost

    - XGBoost outperforms both FT-Transformer and ResNet in baseline scenarios.
    - With effective class imbalance handling methods such as class weight and ROS, DL models can achieve comparable performance to XGBoost, especially in high imbalance ratio datasets under the same techniques in class imbalance problems.

Dataset	IR	None	CW	ROS	SMOTE	ADASYN	RUS	ENN	RENN	ALL KNN
electricity	100	-0.068	0.446	0.160	0.068	0.008	-0.144	-0.155	-0.145	-0.092
electricity	20	-0.259	0.114	0.073	0.063	0.059	-0.064	-0.277	-0.247	-0.255
electricity	10	-0.216	0.002	0.014	0.003	0.004	-0.096	-0.190	-0.194	-0.251
electricity	2	-0.118	-0.100	-0.092	-0.104	-0.098	-0.125	-0.106	-0.084	-0.101
electricity	1	-0.102								
MagicTelescope	100	-0.041	0.279	0.085	-0.107	-0.133	-0.023	-0.064	-0.066	-0.075
MagicTelescope	20	-0.087	0.022	0.064	-0.045	-0.084	-0.035	-0.108	-0.104	-0.088
MagicTelescope	10	-0.053	0.056	0.040	-0.011	-0.039	-0.044	-0.087	-0.090	-0.092
MagicTelescope	2	-0.013	-0.010	-0.001	-0.015	-0.018	-0.034	-0.039	-0.026	-0.036
MagicTelescope	1	-0.011								
GMSC	17	-0.236	0.154	0.071	-0.051	0.160	-0.273	-0.248	-0.260	-0.275
ISCXIDS2012	55	-0.998	-0.120	-0.662	-0.956	-0.714	-0.732	-0.998	-0.998	-0.998

Table 5.7: Subtraction of ResNet - XGBoost in F1 Score for Minority Class, Positive values highlighted in Green

Chapter 6

Conclusion

6.1 Summary

This study presents a comparative analysis of the performance of various class imbalance handling techniques on two DL models (FT-Transformer, ResNet) and one baseline ML model (XGBoost) across various datasets. It highlights the significant impact of class imbalance handling methods on the performance of DL models for tabular data. While XGBoost remains a strong performer when comparing the best F1 scores after applying the best performing class imbalance handling methods for each model, DL models like FT-Transformer and ResNet can achieve competitive performance with the same class imbalance handling techniques.

The key takeaways include:

- **Class Weight and ROS** These methods consistently improve the performance of both FT-Transformer and ResNet models. They are particularly effective in high imbalance ratio scenarios, significantly boosting F1 scores for minority classes.
- **FT-Transformer's Strength in Real-World Data** The FT-Transformer, combined with proper class imbalance handling methods, shows superior performance in real-world datasets, indicating its robustness and applicability in practice. Effective class imbalance handling methods further enhance its performance.
- **XGBoost's Robustness** XGBoost remains a robust choice for handling tabular data with class imbalance, outperforming DL models in baseline scenarios and achieving the best F1 scores with best performing class imbalance handling methods. However, when comparing results where models applied the same class imbalance handling methods, DL models like FT-Transformer

and ResNet can achieve comparable performance and sometimes even superior results under highly imbalanced scenarios.

- **Noise Reduction Techniques** These techniques are generally less effective for DL models and XGBoost, and can sometimes degrade performance. Balancing the sample sizes between classes proves to be a more critical factor for improving model performance.

6.2 Future Work

Future research should focus on the following areas to further enhance the understanding and application of DL models for tabular data with class imbalance.

- **Hybrid Models:** Investigate the potential of combining DL models with XGBoost in class imbalance scenarios to improve performance.
- **Advanced Techniques:** Explore advanced imbalance handling methods like two-phase training and GANs for synthetic data generation.
- **Understanding Effectiveness:** Delve deeper into why certain methods work better for DL models and others for ML models.
- **More Diverse Datasets:** Expand the range of real-world datasets to provide a more comprehensive comparison.

By addressing these areas, future studies can bridge the performance gap between ML and DL models and among DL models themselves, potentially making DL a more practical option for a wider range of tabular data applications.

This research contributes to the field by providing a detailed empirical comparison of class imbalance handling methods applied to DL models for tabular data, highlighting the most effective techniques under various conditions. The study offers practical guidance for selecting appropriate class imbalance handling methods, such as class weight and ROS, to improve DL model performance on imbalanced tabular datasets. It identifies scenarios where DL models, with proper class imbalance handling, can achieve significant improvements in F1 score. The results demonstrate the compelling effectiveness of proper class imbalance handling techniques for DL models, yet they still perform behind XGBoost when comparing the best F1 scores of each model for each dataset. By addressing the gap in research on class imbalance handling in DL models for tabular data, this study stimulates further investigations into DL models for tabular data, contributing to the advancement of the domain.

Bibliography

[1] Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *ArXiv*, abs/1908.07442, 2019.

[2] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35:7499–7519, 2021.

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks : the official journal of the International Neural Network Society*, 106:249–259, 2017.

[4] N. Chawla, K. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *ArXiv*, abs/1106.1813, 2002.

[5] 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*, KDD '16. ACM, August 2016.

[6] Ishan Isaac Dey and Vibhav Pratap. A comparative study of smote, borderline-smote, and adasyn oversampling techniques using different classifiers. *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pages 294–302, 2023.

[7] Alin Dobra. Decision trees. *ACM Inroads*, 10:56 – 56, 2018.

[8] Vaishnavi Nath Dornadula and S. Geetha. Credit card fraud detection using machine learning algorithms. *Procedia Computer Science*, 2019.

[9] Ahmed I. B. Elseddawy, Faten Khalid Karim, Aisha Mohamed Hussein, and Doaa Sami Khafaga. Predictive analysis of diabetes-risk with class imbalance. *Computational Intelligence and Neuroscience*, 2022, 2022.

[10] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.

[11] Yu. V. Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Neural Information Processing Systems*, 2021.

[12] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Neural Information Processing Systems*, 2022.

[13] Fei He, Huamin Yang, Yu Miao, and Rainbow Louis. A cost sensitive and class-imbalance classification method based on neural network for disease diagnosis. *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, pages 7–10, 2016.

[14] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[15] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, 2009.

[16] Sophie Henning, William H. Beluch, Alexander M. Fraser, and Annemarie Friedrich. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2022.

[17] Arlind Kadra, Marius Thomas Lindauer, Frank Hutter, and Josif Grabocka. Regularization is all you need: Simple neural nets can excel on tabular data. *ArXiv*, abs/2106.11189, 2021.

[18] Kaggle. Give me some credit. `https://www.kaggle.com/competitions/GiveMeSomeCredit`. Accessed: 2023-07-08.

[19] Liran Katzir, Gal Elidan, and Ran El-Yaniv. Net-dnf: Effective deep modeling of tabular data. In *International Conference on Learning Representations*, 2021.

[20] Michael Keith. Random forest. *Machine Learning with Regression in Python*, 2021.

[21] Karim Lounici, Katia Meziani, and Benjamin Riu. Muddling label regularization: Deep learning for tabular datasets. *ArXiv*, abs/2106.04462, 2021.

[22] Yang Lu, Yiu ming Cheung, and Yuanyan Tang. Bayes imbalance impact index: A measure of class imbalanced data set for classification problem. *IEEE Transactions on Neural Networks and Learning Systems*, 31:3525–3539, 2019.

[23] Farjad Malik, Simon Wouters, Ruben Cartuyvels, Erfan Ghadery, and Marie-Francine Moens. Two-phase training mitigates class imbalance for camera trap image classification with cnns, 2021.

[24] Farjad Malik, Simon Wouters, Ruben Cartuyvels, Erfan Ghadery, and Marie-Francine Moens. Two-phase training mitigates class imbalance for camera trap image classification with cnns. *ArXiv*, abs/2112.14491, 2021.

[25] Marcus A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. 2003.

[26] Duncan C. McElfresh, Sujay Khandagale, Jonathan Valverde, C. VishakPrasad, Ben Feuer, Chinmay Hegde, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *ArXiv*, abs/2305.02997, 2023.

[27] M.Shraddha, K. Sumedha, and V.Veda Samhitha. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *International Journal For Multidisciplinary Research*, 2023.

[28] Leif E. Peterson. K-nearest neighbor. *Scholarpedia*, 4:1883, 2009.

[29] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *ArXiv*, abs/1909.06312, 2019.

[30] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019.

[31] Qianxi Qiu and Han Liu. Numerical embedding of categorical features in tabular data: A survey. *2023 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 446–451, 2023.

[32] Manisha Saini and Seba Susan. Tackling class imbalance in computer vision: a contemporary review. *Artificial Intelligence Review*, 56:1279–1335, 2023.

[33] Rick Sauber-Cole and Taghi M. Khoshgoftaar. The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data*, 9, 2022.

[34] Ira Shavitt and Eran Segal. Regularization learning networks. In *Neural Information Processing Systems*, 2018.

[35] Ryota Shimizu, Kosuke Asako, Hiroki Ojima, Shohei Morinaga, Mototsugu Hamada, and Tadahiro Kuroda. Balanced mini-batch training for imbalanced image data classification with neural network. *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, pages 27–30, 2018.

[36] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.*, 31:357–374, 2012.

[37] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *ArXiv*, abs/2106.03253, 2021.

[38] Gissel Velarde, Anindya Sudhir, Sanjay Deshmane, Anuj Deshmunkh, Khushboo Sharma, and Vaibhav Joshi. Evaluating xgboost for balanced and imbalanced data: Application to fraud detection. *ArXiv*, abs/2303.15218, 2023.

[39] Priyanka Verma, John G. Breslin, Donna O'Shea, Nakul Mehta, Nitesh Bharot, and Ankit Vidyarthi. Leveraging gametic heredity in oversampling techniques to handle class imbalance for efficient cyberthreat detection in iiot. *IEEE Transactions on Consumer Electronics*, 70:1940–1951, 2024.

[40] Charles Wheelus, Elias Bou-Harb, and Xingquan Zhu. Tackling class imbalance in cyber security datasets. *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 229–232, 2018.

[41] Christopher K. I. Williams. The effect of class imbalance on precision-recall curves. *Neural Computation*, pages 1–5, 2020.

[42] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.*, 2:408–421, 1972.

[43] T. Zhang, Zheyu Zhang, Zhiyuan Fan, Haoyan Luo, Feng Liu, Li-Yu Daisy Liu, Qian Liu, Wei Cao, and Jian Li. Openfe: Automated feature generation with expert-level performance. In *International Conference on Machine Learning*, 2022.

[44] Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *ArXiv*, abs/2210.17128, 2022.

[45] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18:63–77, 2006.

Appendix A

Program Code / Resources

The source code, datasets, and documentation are available at the following GitHub repository:

`https://github.com/eunmi228/dl-class-imbalance-tabular`

Appendix B

Datasets

The following datasets are used in this study:

1. **Electricity**:
    - **Description**: The Electricity dataset contains electricity usage data with features representing various attributes of the usage patterns.
    - **Link**: `https://www.openml.org/d/151`
2. **MagicTelescope**:
    - **Description**: The MagicTelescope dataset includes gamma telescope data to differentiate between high-energy gamma particles and background cosmic rays.
    - **Link**: `https://www.openml.org/d/1120`
3. **Give Me Some Credit**:
    - **Description**: The Give Me Some Credit dataset contains financial history and demographic data of borrowers for credit risk prediction.
    - **Link**: `https://www.kaggle.com/c/GiveMeSomeCredit`
4. **Intrusion Detection Evaluation Dataset (ISCXIDS2012)**:
    - **Description**: The ISCXIDS2012 dataset includes network traffic data capturing various types of network behavior, including normal activities and cyber-attacks.
    - **Link**: `https://www.unb.ca/cic/datasets/ids.html`

Appendix C

Hyperparameter Space

C.1 FT-Transformer Hyperparameter Space

Parameter (Datasets)	Distribution
# Layers	(A,B) UniformInt[1, 5]
Learning Rate	(A,B) LogUniform[3e-5, 3e-4]
Weight Decay	(A,B) LogUniform[1e-6, 1e-3]
# Iterations	(A) 30, (B) 15

Table C.1: FT-Transformer Hyperparameter space. (A) = ISCXIDS2012, (B) = The rest of the datasets. Use default values for feature embedding size, residual dropout, attention dropout, FFN dropout, and FFN factor, which are determined based on the number of layers.

C.2 ResNet Hyperparameter Space

Parameter (Datasets)	**Distribution**
# Layers	(A,B) UniformInt[1, 8]
Layer size	(A,B) [64, 128, 256, 512]
Hidden factor	(A,B) Const(2)
Hidden dropout	(A,B) Uniform[0, 0.5]
Residual dropout	(A,B) Const(0)
Learning Rate	(A,B) LogUniform[1e-5, 1e-2]
Weight Decay	(A,B) LogUniform[1e-6, 1e-3]
# Iterations	(A) 30, (B) 15

Table C.2: ResNet Hyperparameter Space. (A) = ISCXIDS2012, (B) = The rest of the datasets

C.3 XGBoost Hyperparameter Space

Parameter (Datasets)	**Distribution**
Max depth	UniformInt[1, 10]
Learning Rate	LogUniform[1e-4, 1e-0]
Alpha	LogUniform[1e-5, 1e-2]
# Iterations	15

Table C.3: XGBoost Hyperparameter Space.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Master-/Bachelorarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Er- klärung rechtliche Folgen haben wird.

Mannheim, den 31.08.2014 Unterschrift