Novel Algorithm-Level Approaches for Class-Imbalanced Machine Learning

David Twomey

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

March 27, 2023

I, David Twomey, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Machine learning classifiers are designed with the underlying assumption of a roughly balanced number of instances per class. However, in many real-world applications this is far from true. This thesis explores adaptations of neural networks which are robust to class imbalanced datasets, do not involve data manipulation, and are flexible enough to be used with any model architecture or framework. The thesis explores two complementary approaches to the problem of class imbalance. The first exchanges conventional choices of classification loss function, which are fundamentally measures of how far network outputs are from desired ones, for ones that instead primarily register whether outputs are right or wrong. The construction of these novel loss functions involves the concept of an approximated confusion matrix, another use of which is to generate new performance metrics, especially useful for monitoring validation behaviour for imbalanced datasets. The second approach changes the form of the output layer activation function to one with a threshold which can be learned so as to more easily classify the more difficult minority class. These two approaches can be used together or separately, with the combined technique being a promising approach for cases of extreme class imbalance. While the methods are developed primarily for binary classification scenarios, as these are the most numerous in the applications literature, the novel loss functions introduced here are also demonstrated to be extensible to a multi-class scenario.

Impact Statement

Classification is arguably the most common current application area of machine learning. While in many cases, such as recommender systems ("is this movie going to appeal to the subscriber, or not?"), errors are not of much importance, there are other, such as medical diagnosis and screening, where errors, in particular ones in which a positive (in general, having the property that the classifier seeks to identify) instance that is missed (referred to as a false negative or Type-II error) may have serious consequences. However, classifier systems tend to frequently display these types of errors when the number of positive cases is small compared to the number of negative ones. This thesis addresses the problem of classification in such imbalanced scenarios, and presents novel methods that help avoid false negatives in these situations. The novel methods of this thesis are of two kinds: the first changes the representation of the classification problem so that learning is more aggressively directed toward the correct classification of both negative (majority) and positive (minority) examples, and the second helps the system to more easily correctly classify the minority type. Taken together, these novel tools are a significant extension to the current toolkit for classification in machine learning that has also, as has been emphasised above, the potential for a substantial practical value.

Acknowledgements

I would like to thank Dr. Denise Gorse who has taught me so much regarding the skill of research and the coherent presentation of ideas. She has been incredibly generous and supportive in her role as my supervisor and has shown a genuine interest in helping me to achieve my goals.

I would also like to express my sincere gratitude to my parents, aunt Pat, and partner Lorna, who, among many others, have shown me remarkable support and encouragement throughout this process.

Contents

1 Introduction **16**

1.1 Research Motivation 16

1.2 Research Contributions 18

1.3 Related Publications 20

1.4 Thesis Outline 20

2 Background **23**

2.1 The Class Imbalance Problem 23

2.2 Measuring Performance Under Class Imbalance 26

2.2.1 Threshold Based Approach 27

2.2.2 Classification Threshold Invariant Measures 32

2.3 Existing Approaches and Solutions to Imbalanced Classification . . 35

2.3.1 Data-Level Approaches 35

2.3.2 Algorithm-Level Approaches 37

2.3.3 Hybrid Approaches 39

2.4 Summary 40

3 Geometric Mean as a Classification-Focused Training Loss **42**

3.1 Background 42

3.2 A Motivating Preliminary Example 44

3.3 A New Training Metric for Classification 47
3.3.1 The Approximated Confusion Matrix 47
3.3.2 The Approximated G-Mean Loss Function 48
3.4 UCI Dataset Experiment 49
3.4.1 Datasets 50
3.4.2 Experiment Design 50
3.4.3 Results 53
3.5 Comparison to Data-Level Approaches: Pump & Dump 54
3.5.1 Pump & Dump 55
3.5.2 Dataset 56
3.5.3 Experiment Design 58
3.5.4 Results 59
3.6 Discussion 63

4 Classification-Focused Training Metrics: Alternatives & Extension to Multi-Class Scenarios 65

4.1 Alternatives to the Geometric Mean 66
4.1.1 The F1 Score and MCC Loss Functions 67
4.1.2 Image Segmentation 69
4.1.3 IRCAD & HuBMAP Datasets 71
4.1.4 Experiment Setup & Evaluation 74
4.1.5 Results & Discussion 75
4.2 Multi-Class Classification 80
4.2.1 Class Imbalance in the Multi-Class Case 80
4.2.2 Extension of Performance Measures to Multi-Class Classification 81
4.2.3 Construction of Approximated Multi-Class Training Metrics 85

4.2.4 Long-Tailed Recognition 86
4.2.5 CIFAR100-LT Dataset 90
4.2.6 Experiment Setup & Evaluation 90
4.2.7 Results & Discussion 93
4.3 Discussion 102

5 Boosting Performance for Extreme IR: ASTra, a Complementary Approach to Imbalanced Classification 104

5.1 Origin and Derivation of the ASTra Transform 105
5.2 Adaptation of Loss Functions to Accommodate ASTra 110
5.3 Learning the Threshold Value 112
5.4 Data, Experimental Process, and Performance Measurement 115
5.4.1 Data 115
5.4.2 Details of the Experimental Process 116
5.4.3 Performance Measurement 117
5.5 Results 118
5.6 Discussion 122

6 Summary & Conclusions 125

6.1 Discussion of Contributions 125
6.2 Future Work 128
6.3 Concluding Remarks 130

Appendices 131

A Second-Order Derivatives 131

Bibliography 134

List of Figures

1.1 Number of publications on 'class imbalance' by year. Data from [1]. 17

2.1 Binary confusion matrix 27

2.2 Relationship diagram of different performance metrics which can be constructed from the binary confusion matrix. Reproduced from [2] 33

2.3 AUC calculation for the ROC curve-(left) and PR-curve (right). The ROC-curve is composed of the false positive rate (x-axis) and true positive rate (y-axis). The PR-curve is composed of the true positive rate (x-axis) and Precision (y-axis). 33

3.1 Two-dimensional balanced (left) and imbalanced (right) sample binary-classification datasets. 44

3.2 Train and test model performance as a function of training epoch for the balanced (left column) and imbalanced (right column) sample binary-classification datasets. 46

3.3 Underperformance (measured by MCC) of both MSE and BCE relative to GMN, plotted against imbalance ratio (IR). GMN outperforms the competitor cost functions everywhere in the shaded region . . . 53

3.4 Distribution of out-of-sample scores across all random seeds/k-folds (300 total) for GMN, BCE, and MSE cost functions, for the exceptional dataset *abalone-20-vs-8-9-10*. 54

3.5 Test MCC scores on pump and dump data for models trained with BCE and MSE losses, for progressively undersampled train sets, and for GMN loss without undersampling (dotted line). 62

4.1 Types of image segmentation task. The datasets used here are examples of a binary form of semantic segmentation (b). Images reproduced from [3] 69

4.2 Examples of 256 $\times$ 256 tiles and the corresponding masks for the IRCAD and HuBMAP image segmentation datasets. 73

4.3 IRCAD Intersection over Union (IoU) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars. 77

4.4 IRCAD Matthews Correlation Coefficient (MCC) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars. 77

4.5 HuBMAP Intersection over Union (IoU) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars. 78

4.6 HuBMAP Matthews Correlation Coefficient (MCC) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars. 79

4.7 Multi-class classification problem confusion matrix 81

4.8 Label distribution of a long-tailed image dataset. Original image from [4] 87

4.9 CIFAR100-LT train set frequency distributions for allocated class cardinalities for the different target imbalance ratios (IRs), showing also the division of classes into Head, Middle, and Tail class frequencies. 91

4.10 CIFAR100-LT (IR=100): Comparison of average Top-1 Accuracies of the confusion matrix based loss functions used in this chapter (GMN, F1 & MCC losses, displayed in bold) to recent state-of-the art results for this dataset. 97

4.11 Comparison of Top-1 Accuracies on CIFAR100-LT (IR=100) across the second-stage losses considered, for differing train class frequencies. 97

4.12 CIFAR100-LT (IR=100) per-class test Top-1 Accuracy for second-stage model trained with MCC loss compared to per-class test Top-1 Accuracy for baseline model with CE loss and no second-stage training. Red line indicates class count of training dataset (test dataset is balanced). 98

4.13 Comparison of Top-1 Accuracies on CIFAR100-LT (IR=500) across the second-stage losses considered, for differing train class frequencies. 99

5.1 Activation profile for $f(x,b) = (1+be^{-bx})^{(-1/b)}$ for different values of b. At $b = 1$ this is equivalent to the standard sigmoid function. However as b increases, the threshold (i.e. point of maximum slope) *increases*. This setup, the one derived most directly from the Richards equation, works effectively for minority-0 labels but for a conventional minority-1 requires adjustment. 108

5.2 Activation profile for $g(x,b) = 1 - f(-x,b)$ for different values of b. At $b = 1$ this is equivalent to the standard sigmoid function. In this case, as b increases, the threshold (i.e. point of maximum slope) *decreases*. This setup is best for reducing false negatives in the conventional minority-1 setup. 110

5.3 (a) Activation function $ASTra(x,b)$ and (b) its first derivative, for $\tau(b) = 0.5, 0.25, 0.05$. 111

5.4 Loss functions for targets 0 and 1 for (a) $\tau(b) = 0.5$ and (b) $\tau(b) = 0.25$ 113

5.5 z-transforms for $\tau = 0.5, 0.25, 0.05$ 114

5.6 Fold-averaged training behaviour for *skin-4000*, in terms of the (logged) *e-ratio* for each considered method. 121

List of Tables

3.1 Summary of the 30 chosen UCI datasets used for the G-Mean experiment. 51

3.2 Pump & Dump dataset train and test partitions 58

3.3 Test classification results for the original pump and dump dataset. Performance is averaged over 25 runs, with standard deviations shown in brackets. 59

3.4 Pump & Dump datasets resulting from majority-class (non-pump) under-sampling. 61

3.5 Test classification results for the SMOTE versus GMN pump and dump dataset experiment. Performance is averaged over 25 runs, with standard deviations shown in brackets. 64

4.1 IRCAD test results. 76

4.2 HuBMAP test results. 78

4.3 Benchmark CIFAR100-LT (IR = 100) test Top-1 Accuracies. 94

4.4 Top-1 Accuracy on CIFAR100-LT (IR=100) test set, for baseline model plus weight decay (as in [5]), with various choices of subsequent intervention, broken down into regularisation (as in [5]) and second-stage classifier training, the latter including both a loss used in [5] (CE+CB) and three novel loss functions of this thesis (GMN, F1, MCC). "CB": class-balancing [6]; "LDAM" label-distribution-aware margin (LDAM) loss [7]. Results are split vertically by training methodology and horizontally by train class frequency, with standard deviations in brackets, and bolding indicating a win or tie at 5% statistical significance among the second-stage classifier models. 96

4.5 Top-1 Accuracy on CIFAR100-LT (IR=500) test set, for baseline model plus weight decay (as in [5]), with various choices of subsequent intervention, broken down into regularisation (as in [5]) and second-stage classifier training, the latter including both a loss used in [5] (CE+CB) and three novel loss functions of this thesis (GMN, F1, MCC). "CB": class-balancing [6]; "LDAM" label-distribution-aware margin (LDAM) loss [7]. Results are split vertically by training methodology and horizontally by train class frequency, with standard deviations in brackets, and bolding indicating a win or tie at 5% statistical significance among the second-stage classifier models. 100

4.6 Overall (all class frequencies) Matthews Correlation Coefficient (MCC) scores on CIFAR100-LT for the two imbalance ratios (IRs). 101

5.1 Summary of ASTra equations based on target setup. The ASTra formulation varies depending on whether targets are labelled as minority-0 or minority-1. $z(y,\tau)$, $\frac{\partial}{\partial y}z(y,t)$ and $\frac{\partial}{\partial \tau}z(y,\tau)$ remain the same for both. 109

5.2 Datasets used in this work. The shaded bars correspond to the original datasets of [8] and the unshaded bars to our additional minority-undersampled datasets 115

5.3 Mean fold-aggregated test performance (10×5CV), with standard deviations bracketed. Cells with bold type and shading denote winners or ties, where to be deemed a winner a method needs to outperform its competitors with $p \leq 0.05$ 119

Chapter 1

Introduction

This chapter provides an overview of the thesis. First a framing of the topic and the motivations for its research are introduced. Next, the objectives of the research are discussed and a high-level overview of the research and scientific contributions are established. It concludes with a summary of the structure of the subsequent sections.

1.1 Research Motivation

Effective classification with imbalanced data is an important area of research, as high class imbalance is naturally inherent in many real-world applications, such as medical imaging [9], seismology [10] and fraud detection [11]. However, machine learning classifiers are designed with the underlying assumption that the dataset under study has a roughly balanced number of instances per available class, and if trained to minimise a standard loss such as cross entropy will typically fall into the trap of achieving a low error by focusing primarily on the majority class, with many minority examples misclassified. This can give rise to large problems given that the minority class is usually the one of interest, the one in which errors are least acceptable (for example, the misclassification of a tumour image as healthy, the majority class, is likely to be far more problematic than the converse, a false alarm). The issues surrounding imbalanced classification were in the early years of

machine learning less well appreciated, in part due to the dominance of the accuracy performance measure, which can be extremely deceptive in cases of class imbalance (for example, a classifier that predicted all examples to be of the majority type, and was therefore in practical terms useless, would, if the majority comprised 95% of the data, be even so termed "95% accurate").

Performance measures such as the F1-Score [12] and Matthews Correlation Coefficient (MCC) [13], are far less deceptive (both would return a value of zero for the previous example), and the now-widespread use of these in results reporting has clarified the extent of the problem. But there is a difference between acknowledging a problem and solving it. While the number of papers dealing with imbalanced classification has grown very quickly in recent years, as highlighted in Figure 1.1,

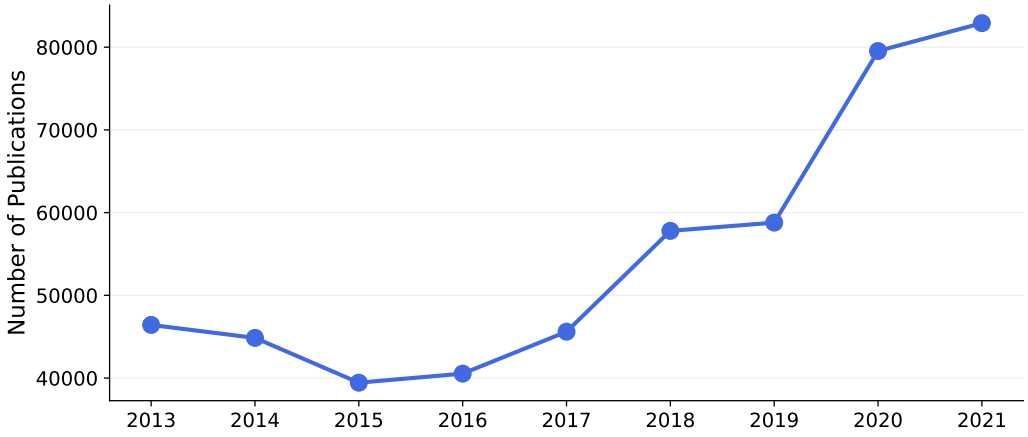

Figure 1.1: Number of publications on 'class imbalance' by year. Data from [1].

the problem is far from solved, and many applications papers handle the problem by largely ignoring it, reporting results on artificially balanced datasets with no guarantee that their results will extend to the imbalanced data found in reality (for example, [10] comments on such issues in relation to the problem of detecting earthquake signals in seismic waveforms). In addition, popular methods for handling class imbalance, such as those that artificially augment the minority class examples,

have been criticised as introducing potentially deceptive bias and/or being difficult to implement, changing the model architecture to one with which the user may be less familiar (a potential barrier to adoption in applications-focused work). Hence, there is room for new work in this field, especially interventions that do not require any manipulation of data or large changes to familiar machine learning models. In addition, there is room for work that tackles the challenge of the extreme class imbalances that can be found, for example, in the area of fraud detection.

In summary, this thesis is motivated by the requirement for methods to address the class imbalance problem that are:

- Flexible, intuitive, easy to implement, and do not require data manipulation or the pre- or post-specification of numerous parameters.
- Able to demonstrate performance improvements on real-world datasets which either match or improve on the current state of the art.
- Extensible to application areas of extreme class imbalance, where the ratio of majority to minority examples is of the order of 100s or 1000s.

1.2 Research Contributions

This thesis explores adaptations of neural networks which are robust to class imbalanced datasets and flexible enough to be used with any model architecture or framework. A key advantage of the approaches taken here is that researchers can develop classifiers without manipulating the data or changing the model architecture, two disadvantages within existing approaches that were noted above. The work presented here considers two complementary approaches to the problem of class imbalance. The first approach replaces traditional classification loss functions, like cross-entropy, which primarily focus on measuring the error between network out-

puts and desired outputs, with loss functions that mainly determine if the outputs are correct or incorrect based on a predefined threshold (typically 0.5 for classifiers with outputs between 0 and 1). The construction of these novel loss functions involves the concept of an approximated confusion matrix; an additional use of this matrix is to generate metrics for performance measurement that give a more complete picture of the progress of training for imbalanced datasets, and are especially valuable for monitoring validation behaviour. The second approach changes the form of the output layer activation function to one with a trainable asymmetry, so that the classification threshold is no longer necessarily 0.5 but can now be learned so as to favour the classification of the minority class, examples of which might otherwise tend to be misclassified as belonging to the majority. These two approaches can be used together or separately, with a combined technique being a promising approach for instances of extreme class imbalance, where the ratio of negative to positive examples is of the order of 1000s. These two complementary methods are developed primarily for binary classification scenarios, as these are the most numerous in the applications literature; however, the novel loss functions introduced here are also demonstrated to be extensible to a multi-class scenario.

In summary, the contributions of this thesis are:

- Novel classification-focused loss functions that are parameter-free and demonstrated to be effective across a range of problems and domains.
- The use of an approximated confusion matrix as a source of valuable metrics for monitoring training behaviour for imbalanced datasets.
- Demonstration that these loss functions can additionally be generalised for effective use in multi-class scenarios.
- A novel output layer activation function that can complement the use of the

above loss functions by making it easier to classify minority examples.

1.3 Related Publications

Parts of this thesis have been published in the European Symposium on Artificial Neural Networks (ESANN):

D. Twomey and D.Gorse (2018), A neural-network cost function for highly class-imbalanced data sets, *Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, April 25-27, 2018,

and submitted to the International Conference on Artificial Neural Networks (ICANN):

D. Twomey and D.Gorse (2022), ASTra: A Novel Algorithm-Level Approach to Imbalanced Classification, *Proceedings of the 31st International Conference on Artificial Neural Networks*, Bristol, United Kingdom, September 6-9, 2022.

1.4 Thesis Outline

The structure of the remaining parts of this thesis is as follows:

- **Chapter 2** (Background) provides the necessary background to this work and a survey of related literature. The background section first introduces the problem of class imbalance, then covers the substantial topic of how performance should best be measured in situations of class imbalance. This chapter concludes with a literature survey reviewing existing approaches to the problem of class imbalance, gradually focusing on the main area of this thesis: algorithm-level adaptations which focus on the loss objective and final-layer activation.

- **Chapter 3** (Geometric Mean as a Classification-Focused Training Loss) explains how, by looking at a confusion matrix of model predictions, it is possible to understand the difficulties in using standard (regression-focused) loss functions as a proxy for training classifiers. This motivates the presentation of an approximated form of the confusion matrix, from which differentiable forms of well known class-imbalance-robust performance measures such as the geometric mean (G-Mean) can be defined. The outperformance of this new approximated G-Mean loss against standard loss functions is demonstrated on a wide set of common class-imbalanced UCI datasets, and on a challenging real-world dataset relating to a specific, and regularly occurring, form of manipulation of cryptocurrency markets.

- **Chapter 4** (Classification-Focused Training Metrics: Alternatives & Extension to Multi-Class Scenarios) considers alternatives to the G-mean loss of the previous chapter, introducing further novel loss functions based on the F1-Score and the MCC. These, together with the G-Mean loss, are then applied to the problem of image segmentation (classification at the pixel-level), for two publically available medical image datasets, with results that are at least competitive with the state of the art. The use of novel loss functions is then extended to the multi-class domain, specifically the problem of Long Tailed Recognition (LTR), with results in this instance significantly better than the current state of the art.

- **Chapter 5** (Boosting Performance for Extreme IR: ASTra, a Complementary Approach to Imbalanced Classification) introduces a novel output layer activation function to facilitate the classification of minority examples in scenarios of high class imbalance. ASTra is applied to the most imbalanced datasets from

a recent extensive study of ensemble classifiers and achieves performances comparable to the state of the art, despite restricting the models to very simple single hidden layer networks with a small number of neurons (between three and 12 hidden neurons). ASTra can be used on its own; however it is shown that its combination with the G-Mean from Chapter 3 can provide performance benefits on the most extreme class-imbalanced problems.

- **Chapter 6** (Summary & Conclusions) discusses the research presented in this thesis, reflecting on the aims outlined in the first chapter and the solutions subsequently found to address them. Future research possibilities are provided, along with concluding remarks.

Chapter 2

Background

This chapter introduces the necessary domain and technical background relevant to themes present in this research. This begins with an introduction to the class imbalance problem and why it is of such importance in real-world settings, then introduces necessary machine learning concepts used in subsequent chapters, with a focus on classification under class imbalance. Lastly, we explore the literature for clues as to how others have attempted to solve the problem categorising approaches into that of data-level, algorithm-level, and hybrid approaches.

2.1 The Class Imbalance Problem

Supervised learning methods require labelled training data, and in a classification task, each data sample belongs to a known class or category [14]. In a binary classification setup, where data samples belong to one of two classes, class imbalance is said to occur when one class, the minority class, contains significantly fewer samples than the other class, known as the majority class. Typically, the minority group is the class of interest (i.e. the positive class) [15]. This form of imbalance is referred to as *between-class* imbalance, which is measured as a ratio between the majority and minority class. For example, a dataset with 100,000 examples with just 1,000 labelled as the minority class is described as having a class imbalance of 100:1.

It is not uncommon for datasets to contain class imbalances of the order of 100:1, 1,000:1, or even 10,000:1, where in each case, one class severely out-represents another [16, 17].

The class imbalance problem can be generalised to multi-class data, where the number of majority classes or minority classes may be one or more [18]. This chapter focuses primarily on class imbalance in the two-class binary setting, with a more specific examination of multi-class class imbalance found in Chapter 4. However, for either the binary (number of classes K=2) or multi-class (K $>$ 2) cases, class imbalance is generally represented as the maximum between-class imbalance level, defined by the Imbalance Ratio (IR) ratio, given by

$$IR = \frac{\max_i |C_i|}{\min_i |C_i|},\ i = 1..K, \tag{2.1}$$

where C_i is the set of all examples in class i, and $\max_i |C_i|$ and $\min_i |C_i|$ return the maximum and minimum class size over all i, respectively [15].

Class-imbalanced problems naturally arise in many applications, examples of which include the medical diagnosis task of detecting disease [19, 20, 21, 22, 23], where the majority of patients are healthy but detecting those with the disease is of greater interest, and fraud detection [24, 25, 26] where the detection of rare, but costly, fraudulent transactions in a payments network is desirable. Other examples include computer security [27, 28], fault detection [29, 30], oil-spill detection [16], and earthquake prediction [31, 32]. Furthermore, new class-imbalanced problems appear all the time, for example the recent rise in fake job descriptions appearing on online social networks [33].

When the imbalance is a result of naturally occurring frequencies of data (e.g. medical diagnoses where the majority of patients are healthy), this is referred to as *intrinsic* imbalance. In contrast, *extrinsic* imbalance occurs when class imbalance

is introduced through external factors such as data collection or storage procedures [34].

An imbalance in the data distribution, whether intrinsic or extrinsic, can present challenges for classification. It is often the case that when class imbalance exists within training, models will typically learn to over-classify the majority group due to its increased prior probability. As a result, the instances belonging to the minority group are more often misclassified than those belonging to the majority group increasing the chance of Type-II errors also known as *false negatives*. Furthermore, some evaluation metrics, such as accuracy, may produce misleading results that indicate good performance under such circumstances: a naive model which always predicts the majority (negative) class on a dataset with only one positive sample for every 100 will achieve a 99% accuracy, even though it has no real discretionary power.

This having been said, some studies have asserted that class imbalance is not always the primary factor that hinders learning, showing that for certain class-imbalanced datasets the minority concept is accurately learned with little disturbance from the imbalance [35, 36]. Indeed, some conclude that the primary determining factor of classification deterioration is in fact dataset complexity, which, in turn, is amplified by the addition of relative imbalance [37]. Data complexity is a broad term that comprises issues such as overlapping, lack of representative data, small disjuncts and others [34]. Japkowicz [38] examined the effect of class imbalance on artificial datasets with various levels of complexity, training set size and degrees of imbalance and concluded that sensitivity to imbalance increases as the complexity increases, and that non-complex, linearly separable problems are unaffected by any level of class imbalance. In fact, [39] argue that good results can be obtained, regardless of class disproportion, if all groups are well represented and come from

non-overlapping distributions.

It should also be noted that the number of minority samples is generally more important than class imbalance per se. In other words, it is important to consider the difference between *relative* imbalance and *imbalance due to rare instances*, (also known as '*absolute rarity*') [34]. In some domains, there exists a high class imbalance but a reasonably large set of examples, in absolute terms, of the minority class. This is referred to as *relative* imbalance. Unfortunately however, in other domains, there may be a genuine lack of data due to the low frequency with which events occur (e.g. detecting earthquakes or oil spills [16]), yet it is the minority class which remains the class of interest. This is referred to as the *imbalance due to rare instances*. For example, given a dataset with 10,000 examples and a 100:1 between-class imbalance, you would expect to see 100 minority class examples; clearly, the majority class dominates the minority class. However, if you were to double the sample size of the dataset, (for example in a biomedical study doubling the number of trial participants), and the distribution remained the same, you would have 200 examples of the minority class but the minority class is not now necessarily rare in its own right, but only relative to the majority class. There is again an emphasis on the quality of representation, and cases where the minority class is rare or under-represented are where performance is more likely to be compromised [40]. [41] discusses the difficulties of learning from these rare events and explores various machine learning techniques for addressing these challenges.

2.2 Measuring Performance Under Class Imbalance

The are two fundamental approaches to measuring classifier performance, applicable both to balanced and imbalanced scenarios. The first is a threshold-based approach, discussed in subsection 2.2.1, which revolves around a so-called confusion matrix

(in some domains referred to as a contingency matrix), in which the count in a given matrix cell is ultimately dependent on the choice of a threshold to which the output of a classifier can be compared (for simplicity, we will consider in this section only binary classification, the primary focus of this thesis). The second approach to performance measurement allows this threshold to be varied, looking at the performance of the model over all thresholds, in this way deriving a classification threshold invariant measure, the best-known such measure being the Area Under the ROC Curve (AUROC), which will be discussed in subsection 2.2.2.

2.2.1 Threshold Based Approach

The confusion matrix, referred to above, is defined as the matrix that tabulates predicted vs. actual class instances. An element of the confusion matrix at row i and column j represents the number of instances for which the predicted class is j and the actual class is i. The confusion matrix in this way shows the ways in which the classification model is 'confused' when it makes predictions [42].

Figure 2.1: Binary confusion matrix

		Predicted Class		
		Negative	Positive	Total
Actual Class	Negative	TN	FP	$TN+FP$
	Positive	FN	TP	$FN+TP$
	Total	$TN+FN$	$FP+TP$	N

For a binary problem, in which columns refer to the prediction labels and rows to the actual labels (targets), the confusion matrix (CM) can be represented by four quadrants, as in Figure 2.1, where TP denotes the *true positives* (correctly predicted positive examples), TN denotes the *true negatives* (correctly predicted negative examples), FN denotes the *false negatives* (positive instances predicted to be negative), and FP the *false positives* (negative instances predicted to be positive).

Often one of the best descriptors of classification performance that can be derived from the confusion matrix is the matrix itself: it provides an insight not only into the errors being made by a classifier, but more importantly the type of the resulting errors. However, it is often more helpful to be able to describe a model's performance via a single performance measure, and it is possible to derive a wide range of performance measures from the confusion matrix. This said, a single metric can never even in principle be an ideal judge of the performance of an algorithm and therefore several complementary measures should be taken into account. Furthermore, in the presence of class imbalance, some metrics, such as Accuracy, may provide a misleading picture [42], as will be emphasised below.

One of the most commonly used metrics used when evaluating classification results is the *Accuracy* (Equation 2.2) and *Error Rate* (Equation 2.3):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

$$Error\ Rate = 1 - Accuracy \tag{2.3}$$

These measures provide a simple way of describing a classifier's performance on a given dataset. However, on class imbalanced datasets, both can be dominated by the majority group (i.e. the negative class) and are therefore no longer reliable measures, because they provide an over-optimistic estimation of the classifier ability on the majority class [43]. For example, on a dataset whose positive group distribution is just 1% of the dataset, a naive classifier can achieve a 99% Accuracy score simply by labelling all examples as negative. Of course, such a model provides no real discretionary power and thus no real value [15]. In other words, the Accuracy and Error Rate metrics in this case do not provide adequate information on a classifier's functionality with respect to the type of classification required. Many works exist

which examine the ineffectiveness of Accuracy for class-imbalanced problems, for example [44, 45, 46], but the fundamental issue can be explained by inspecting the confusion matrix (Figure 2.1). The first row represents negative instances of the dataset and the last row represents positive instances. Therefore, any metric that uses values from both rows will be inherently sensitive to imbalances.[1] As can be seen from Equation 2.2, Accuracy uses both rows' information; therefore, as the class distribution varies, measures of the performance will change even though the underlying fundamental performance of the classifier does not [34]. This can be very problematic when comparing the performance of different learning algorithms over different datasets because of the inconsistency of performance representation.

Evaluation metrics commonly used in the research community to provide better assessments of performance on imbalanced data problems include the *Precision*, also known as the Positive Predictive Value (PPV) and *Recall*, also known as the True Positive Rate (TPR):

$$Precision = PPV = \frac{TP}{TP+FP} \tag{2.4}$$

$$Recall = TPR = \frac{TP}{TP+FN} \tag{2.5}$$

However, taken individually, Precision or Recall are not necessarily a good indicator of performance since one can arbitrarily improve the Precision if there is no constraint on the Recall (and vice-versa). This said, between them they capture some information about the rates and kinds of errors made. However, these two measures and their combinations, such as the *F-measure* defined below, only focus on the positive examples and predictions, neither of them capturing any information about

[1] Some choose to represent the confusion matrix differently, for example with rows representing the predictions and columns the actual classes, or swapping the Negative & Positive positions, but this does not change anything provided the quadrants remain correctly labelled.

how well the model handles negative cases since neither of them take into account the True Negatives (TNs).

Despite the above caveats, the F-measure [12], defined by

$$F\text{-}Measure = \frac{(1+\beta^2) \times Recall \times Precision}{\beta^2 \times Recall + Precision}, \tag{2.6}$$

is a well-known, and commonly recommended, metric for evaluating performance under class imbalance, where the coefficient β can be adjusted to reflect the relative importance of Precision vs. Recall. Compared to measures like Error Rate in binary classification, the F-measure enforces a better balance between performance on the minority and the majority class, respectively, though again it should be emphasised that it does not address performance on negative examples. It is possible to generalize the F-measure to a weighted harmonic average of the Precision and Recall, but for the sake of simplicity, it is common to stick to the unweighted mean, which is referred to as the *F1-Score* (where $\beta = 1$).

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.7}$$

In order to address performance on the negative class, we need other metrics such as the *Specificity*, also known as the True Negative Rate (TNR),

$$Specificity = TNR = \frac{TN}{TN + FP}, \tag{2.8}$$

and the *Negative Predictive Value*,

$$NPV = \frac{TN}{TN + FN}, \tag{2.9}$$

A key metric used in the work of this thesis, the *G-Mean*, which evaluates the

degree of inductive bias in terms of a ratio of positive accuracy and negative accuracy [34], combines measures that evaluate performance on both positive and negative examples, namely Recall and Specificity,

$$\begin{aligned} \textit{G-Mean} &= \sqrt{\textit{Recall} \times \textit{Specificity}} \\ &= \sqrt{TPR \times TNR} \\ &= \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \end{aligned} \tag{2.10}$$

However, the *Matthews Correlation Coefficient* (MCC) combines information about positive and negative performance in a more complete way, and as such is considered one of the most reliable metrics for measuring performance on class-imbalanced datasets. It is a reliable statistical measure which only produces a high score if predictions give good results in all of the four confusion matrix (see Figure 2.1) categories, proportionally to both the number of the positive elements and the number of the negative elements in the dataset [47]. The MCC, which is a special case of the ϕ *(Phi) Coefficient* [48], was originally developed by Matthews in 1975 for a comparison of chemical structures [13] and re-proposed by Baldi et al. [49] as a standard performance metric for machine learning with natural extensions to the multi-class setting. The MCC is in essence a correlation coefficient: an MCC score of +1 represents a perfect prediction, 0 either no better than random or all assigned to one class, and -1 an inverse prediction.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \tag{2.11}$$

Both the F1-measure and the MCC have been widely used in the classification literature (the use of the G-Mean has been in comparison quite limited). The MCC

has clear advantages over the F1 Measure. Firstly, in binary classification, the MCC is invariant to positive and negative class labels being swapped, whereas the F1 produces different results. While this issue can be overcome by extending a macro/micro averaging procedure to make the F1 invariant to class swapping [50], the procedure is biased and is still far from being accepted as standard practice by the community [47, 51]. Secondly, unlike the MCC, the F1 score is independent of the number of samples correctly classified as negative, since it does not examine the number of true negatives (TNs). This obvious drawback/conceptual flaw has been highlighted by [52], but the F1 still remains as one of the most widespread metrics among researchers [47].

Aside from the above-discussed, many other metrics can be constructed from the confusion matrix. This number is vast; as such, some less common or peripheral metrics have been omitted from this discussion in the interest of space. In addition, the same underlying metric is often referred to by different names across different subject-domains. For completeness, Figure 2.2 illustrates the relationship between different metrics, including some other metrics not covered previously in this chapter.

2.2.2 Classification Threshold Invariant Measures

The above subsection described approaches to measuring binary classification success that were ultimately dependent on the choice of a threshold, with model outputs above this threshold being assigned to the positive class, and conversely for the negative class. While it is possible to consider threshold-modifying models, as in Chapter 5, which allow the choice of a threshold to be largely circumvented, it is also possible to construct performance metrics that are entirely independent of the threshold, though these may not always be appropriate for use with imbalanced data, as was noted in [53].

Receiver Operator Curves (ROC) and Precision-Recall (PR) curves provide

		Predicted condition			
	Total population = P + N	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) = TPR + TNR − 1	Prevalence threshold (PT) = $\frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP)**, hit	**False negative (FN)**, type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power = $\frac{TP}{P}$ = 1 − FNR	False negative rate (FNR), miss rate = $\frac{FN}{P}$ = 1 − TPR
	Negative (N)	**False positive (FP)**, type I error, false alarm, overestimation	**True negative (TN)**, correct rejection	False positive rate (FPR), probability of false alarm, fall-out = $\frac{FP}{N}$ = 1 − TNR	True negative rate (TNR), specificity (SPC), selectivity = $\frac{TN}{N}$ = 1 − FPR
	Prevalence = $\frac{P}{P+N}$	Positive predictive value (PPV), precision = $\frac{TP}{PP}$ = 1 − FDR	False omission rate (FOR) = $\frac{FN}{PN}$ = 1 − NPV	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$	Negative likelihood ratio (LR−) = $\frac{FNR}{TNR}$
	Accuracy (ACC) = $\frac{TP+TN}{P+N}$	False discovery rate (FDR) = $\frac{FP}{PP}$ = 1 − PPV	Negative predictive value (NPV) = $\frac{TN}{PN}$ = 1 − FOR	Markedness (MK), deltaP (Δp) = PPV + NPV − 1	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$
	Balanced accuracy (BA) = $\frac{TPR+TNR}{2}$	F_1 score = $\frac{2PPV \times TPR}{PPV+TPR}$ = $\frac{2TP}{2TP+FP+FN}$	Fowlkes–Mallows index (FM) = $\sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) = $\sqrt{TPR \times TNR \times PPV \times NPV}$ − $\sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index = $\frac{TP}{TP+FN+FP}$

Figure 2.2: Relationship diagram of different performance metrics which can be constructed from the binary confusion matrix. Reproduced from [2]

such a diagnostic tool for binary classification models. They are constructed by adjusting the decision threshold, (which by default for any classifier with outputs in the range $\in [0, 1]$ would be set at 0.5, halfway between each outcome) between 0 and 1. The figure below (Figure 2.3) shows both a ROC and PR curve:

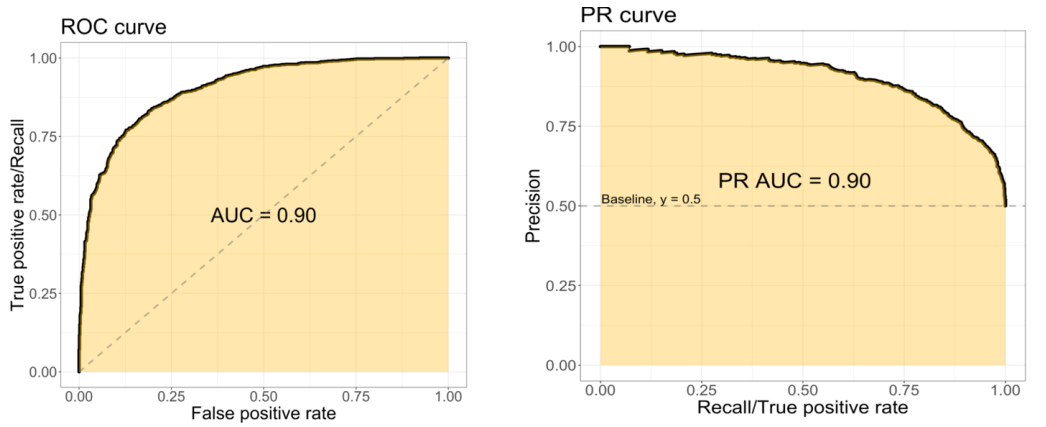

Figure 2.3: AUC calculation for the ROC curve-(left) and PR-curve (right). The ROC-curve is composed of the false positive rate (x-axis) and true positive rate (y-axis). The PR-curve is composed of the true positive rate (x-axis) and Precision (y-axis).

The ROC curve, the most common, allows a trade-off between the TPR and

FPR: changing the threshold of classification will change the balance of predictions towards improving the TPR, at the expense of the FPR, or vice versa, and the ROC curve is constructed from the collection of points this generated. However, it can be challenging to compare two or more classifiers based on their ROC curves. Instead the "Area Under the Curve" (AUC) can be calculated to give a single score for a classifier across all classification threshold values. This is called the ROC Area Under the Curve or AUC_{ROC} (sometimes AUROC or ROCAUC). The score is between 0 (inverse prediction) and 1 (a perfect classifier), with a value of 0.5, appearing as a diagonal line on the TPR-FPR plot, corresponding to a random output, and thus to the worst performance one might expect to usually observe. The AUC_{ROC} can be interpreted as the probability that the scores given by a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [54].

Equivalently to the ROC case, the Precision-Recall AUC can be used as a summary statistic for the PR curve. AUC_{PR} does not have a probabilistic interpretation as the AUC_{ROC} does. The AUC_{PR} of the random classifier varies with the prevalence of the positive class, and its expected value is close to the proportion of positive instances in the test set [55]. In the case of severe class imbalance these curves may tell different stories, given that the PR curve focuses on the minority class alone, whereas the ROC curve covers both classes. Davis & Goadrich [56] argue that when dealing with highly skewed datasets PR curves are more informative, and better able to distinguish models, than ROC curves, since Precision is directly influenced by class imbalance, though noting that the calculation of AUC for PR curves is necessarily more complex and time-consuming, given that the use of methods suitable for ROC curves may for PR curves be prone to yielding an overly-optimistic estimate of performance.

However, there have been some arguments raised against the use of either

AUC_{ROC} or AUC_{PR}. When taking classification costs into account, the model must be fine-tuned to achieve optimal results, ultimately converging to a single point on both the ROC and PR curves. Thus, a classifier with a higher AUC, which considers the whole ROC or PR curve, does not necessarily give a better result [57]. Some researchers suggest incorporating misclassification costs in conjunction with ROC curves for model selection. For instance, [58] argue that optimal solutions can be found as cut points on the ROC curves using tangents with specific slopes determined by misclassification costs. However, this approach is not widely adopted, and most academic studies examining classifier performance overlook the importance of misclassification costs. In practice, these costs can be significant, with false negatives often being more concerning than false positives. Consequently, basing judgments on AUCs might lead to misleading evaluations of classification performance and unwise choices of models in real-world applications.

2.3 Existing Approaches and Solutions to Imbalanced Classification

Having discussed the various ways in which one can measure performance reliably for imbalanced data classification tasks, the focus now shifts to ways in which researchers have attempted to address and alleviate the problem. Hundreds of algorithms have been proposed which can be broadly grouped into three categories: data-level, algorithm-level, and hybrid approaches [15].

2.3.1 Data-Level Approaches

Data-level techniques aim to reduce the level of class imbalance through some form of pre-processing which modify the training class distribution, the most common data-level technique being re-sampling [4], in the form of either under-sampling or

over-sampling. Under-sampling methods eliminate the potential harm of skewed data distributions by discarding a number of instances from the majority class, which also has the benefit of reducing the overall size of the training data, consequently increasing the speed of classifier learning. The simplest, yet most effective, method is Random Under-Sampling (RUS) which selects the instances to be removed in a random fashion [59]. In contrast, over-sampling methods create and add new minority class instances into training, with random over-sampling being the simplest of these approaches; this randomly duplicates a certain number of samples from the minority class and adds these back into the original data set [60].

Both under-sampling and over-sampling are very easy to implement; however they introduce different trade-offs. Under-sampling can lead to a loss of important information, while over-sampling can lead to over-fitting on the minority class. Over-sampling, aside from causing and increase in training time due to an enlarged training set, can suffer from over-fitting.

More complex over-sampling methods such as the Synthetic Minority Over-sampling Technique (SMOTE) [61] have been developed to alleviate issues with over-fitting. SMOTE and its variants [62, 63, 64] create artificial data based on the similarities between existing minority samples, and have demonstrated a degree of success on several applications [61, 65]. However, SMOTE draws criticism for ignoring the original distribution of the dataset, leading to the possibility of erroneously expanding the minority region producing a high variance in results and suffering from over-generalisation [34, 66]. Other researchers have attempted to produce alternatives to SMOTE that can generate high-quality synthetic instances, though these tend to be domain-specific in application. For example, [67] used GANs to generate synthetic seismic waveform data, while more recently [68] used an approach they called ”DeepSMOTE” to augment several benchmark image datasets.

Other researchers have explored alternatives to SMOTE for generating high-quality synthetic instances, although these approaches are often domain-specific in their application. For example, [67] used GANs to generate synthetic seismic waveform data, while more recently, [68] employed an approach called "DeepSMOTE" to augment several benchmark image datasets. In addition, [69] proposed using Variational Autoencoders (VAEs) for synthetic data generation in the context of imbalanced learning.

2.3.2 Algorithm-Level Approaches

Algorithm-level approaches involve performing specific modifications on the algorithms dedicated to improving a given classifier. The most common modification involves cost-sensitive learning, which seeks to re-balance classes by adjusting the loss for different classes during training [4]. The method tackles the imbalanced learning problem by using different cost matrices that describe the costs of misclassifying any particular data example. Some argue that there is a strong relationship between cost-sensitive learning and learning from imbalanced data, and that consequently the theoretical foundations and algorithms for cost-sensitive methods can naturally be applied to imbalanced learning problems [70, 41, 45]. Cost sensitive learning is often more computationally efficient than data-level approaches, but remains less popular [8]. [71] conclude that the unpopularity of cost sensitive learning is caused by both the difficulty in determining the appropriate misclassification costs *a priori*, as this requires domain knowledge, and the lack of support provided in many machine learning packages.

The form of cost-sensitive adaptation that best matches the approach employed in Chapters 3 and 4 is to replace a loss function based on distance of an output from its target with one based on classification success. For class-imbalanced datasets, such a loss function may be able to avoid the trap of decreasing the loss by merely

driving already-correct majority outputs closer and closer to their target values. The earliest (known) work dates back to 2013, when [72] proposed a loss function based on the F1-score and used this for the enhancement of document images. Several works appeared between 2017-18, the current author's proposal of an approximated G-Mean [73] (see Chapter 3) a new F1-score based loss function [74] and the Focal Loss [75]. In 2019, two further variants of the F1-Score were proposed, the first [76] used to train a CNN to classify emotions in tweets, and the second [77] proposed specifically for linear models, applied to synthetic and image data. Also in 2019, [78] used a CNN and a multi-class variant of the F1-score to perform cell segmentation. Most recently, in 2021, [79] used a CNN, trained this time with an approximated MCC, to classify skin lesions. Notably, most of the above work has used loss functions derived from the F1-Score. However, as already discussed in subsection 2.2, this measure has been criticised for insufficiently addressing false negatives, and thus may not be optimal in areas such as medical diagnosis, where the cost of these category of errors can be very high.

An alternative to the proposal of a new loss is the adaptation of an existing one to incorporate class weights. The Focal Loss (FL), proposed by [75], and discussed in more detail in Chapter 4, reformulates the cross-entropy (CE) loss to dynamically down-weight the correctly classified samples. [80] propose a modification of the FL, down-weighting samples with either very large gradients or small gradients, as samples with either very large or very small gradients tend to be well classified. [6] introduce an approach, which they call the class-balanced loss (CB), which assigns costs to different classes based on the number of samples within the neighbouring region rather than the number of samples for each class. [7] propose a label-distribution-aware margin loss (LDAM), a regularisation that extends the soft-margin loss by encouraging the minority classes to have larger margins.

Threshold adjustment, also called threshold moving or post-scaling, is a post-processing approach (and thus to be clearly distinguished from the ASTra method proposed in Chapter 5) which adjusts the decision threshold of a classifier in the out-of-sample stage. The most basic version simply divides the network outputs for each class by its estimated prior probability [81], while similar ideas have been incorporated into deep learning in [82].

2.3.3 Hybrid Approaches

Hybrid methods typically combine one or more of the above-mentioned types of approach in various ways to handle the class imbalance problem. In addition, ensemble architectures which support class weighting (an algorithm-level intervention) can be considered to be a hybrid approach when augmented by a data-level technique such as undersampling.

Beginning with non-ensemble hybrid approaches, [83] blend elements of under-sampling, over-sampling, and a novel loss function they name the class rectification loss (CRL) to address class imbalance on computer vision datasets, obtaining state-of-the-art results on several large multi-label, multi-class datasets with extreme levels of class imbalance. [84] introduce the large margin local embedding (LMLE) method, which involves combining a newly-designed quintuplet sampling method and novel triple-header hinge loss function; they demonstrate this combination can learn an effective representation of minority classes on imbalanced image data. [85] propose an improved version of the LMLE methods using clustering techniques they call cluster-based large margin local embedding (CLMLE).

Turning now to ensemble hybrid approaches, these include RUSBoost [86] and the recently proposed HD-Ensemble [8]. RUSBoost is a boosted ensemble of classifiers trained on under-sampled data. [87] propose an improved form of RUSBoost, replacing random undersampling with evolutionary under-sampling,

which they call EUSBoost. The HD-Ensemble of [8] is an emphatically hybrid method, being a combination of ensemble learners with a hybrid data-level approach; the members of the ensemble are designed to be diverse and complementary to each other, allowing their combination with more aggressive strategies of class balancing. However, ensembles require increased computational resources and an increase in training duration, which often make ensembling an infeasible approach for most practical applications, particularly for deep learning. Indeed, only a limited number of computational packages [88, 89, 90] have incorporated the idea of ensemble learning. Furthermore, and perhaps more importantly, ensemble methods per se are also not designed specifically for situations of class imbalance and can still suffer from the over-fitting to the majority class and a loss of generalisation.

2.4 Summary

This chapter has introduced the class imbalance problem, methods to measure model performance under conditions of class imbalance, and a range of existing solutions to the class imbalance problem, encompassing data-level, algorithm-level, and hybrid solutions. Of these three approaches, the algorithm-level appears most promising on the basis that it neither discards data of potential value, nor does it create data that could be potentially misrepresentative of the problem.

Of the various algorithm-level interventions, one of the most elegant, and potentially fruitful, approaches is to create novel loss functions that specifically target the problematic imbalance, by being based on measures that are sensitive to class imbalance. Chapters 3 & 4 of this thesis will primarily create novel loss functions from measures that address both Type I and Type II errors, such as the G-Mean (Chapter 3) and the MCC (Chapter 4), though work with novel loss functions will also give consideration to an approximated F1-measure as the basis for a loss

function, due to the popularity in the literature of the F1 as a performance measure.

However, it should be noted that the existing algorithm-level approaches described in this chapter are not the only possibilities, and that another algorithm-level approach could be to modify the way that components of the classifier behave, for example changing the form of the activation function for the output layer of a neural network to facilitate minority classification, an approach that will be explored in Chapter 5.

Chapter 3

Geometric Mean as a Classification-Focused Training Loss

This chapter introduces a new loss function for the training of a neural network classifier in a class-imbalanced classification setting. This function, based on a differentiable approximation to the G-Mean, represents a balance of sensitivity (recall) and specificity and is thus well suited to problems where other loss functions such as the mean squared error (MSE) and binary cross entropy (BCE) are prone to over-predicting the majority class. Using an approximated confusion matrix in which all elements are differentiable, we show that this new loss function has a relatively simple derivative form.

3.1 Background

As discussed in Chapter 2, datasets with high class imbalance are extremely common in many real-world machine learning problems. When working with these datasets, the training set is composed of a few positive examples and a large number of negative examples. Standard classification algorithms struggle to learn on these datasets since their calibration is typically based on the minimisation of an MSE or BCE loss. Often, the trivial solution consists in assigning the majority label to any

test query, leading to a high performance from an accuracy perspective but missing the (positive) examples of interest. For this reason, simple accuracy is not ideal as a performance measure, since it does not distinguish between a classifier which is simply always predicting the majority class and one which has learned actual discriminatory power on the class problem.

Rather than using a conventional MSE or BCE loss, it would be desirable to optimise one of the more class-imbalance robust measures directly, as these provide a more holistic evaluation over the entire data; for example, one could choose to use the G-Mean (Equation 2.10), the Matthews Correlation Coefficient (Equation 2.11), or the F-Score (Equation 2.7) introduced in Chapter 2. However, when derived from a usually-constructed confusion matrix in which each matrix element gives a simple count of examples, these measures are not differentiable, so it is necessary move from a conventional confusion matrix to an approximated form in which the accumulated entries are essentially probabilities rather than counts. This is not a difficult construction and has potential value even outside of the context of novel loss functions, as measures derived from it can give more finely-tuned information about training and validation progress.

When using the approximated confusion matrix to create a novel loss function, however, another issue needs to be addressed, which is that measures derived from it in their raw forms are *non-decomposable* over the dataset and cannot be expressed as a sum of errors on individual data points [91]. As such, others who have considered loss functions that might be better measures of classification performance have resorted to surrogates of their ideal measures like the hinge loss (in Support Vector Machines), logistic loss (in logistic regression), or the exponential loss (in boosting) all of which are convex functions. The constructions of this chapter, and elsewhere in this thesis, stand in contrast to this, as they provide not surrogates but direct

approximations. The issue of non-decomposability is not a substantial one, as it can be overcome by doing two passes through each batch.

3.2 A Motivating Preliminary Example

The aim of this simple example is to demonstrate the difficulties created by class imbalance. Two sample datasets are constructed with different levels of class imbalance by sampling from two clusters of points normally distributed about vertices of an 2-dimensional hypercube. The first, balanced, dataset has an equal distribution between majority and minority classes. The second, imbalanced dataset is constructed to have a class imbalance of 100:1 (i.e. IR = 100). Figure 3.1 plots the two datasets side-by-side. It should be noted that there is a degree of overlap between the datasets, so even in the balanced case it would not be possible to separate the examples perfectly and achieve an MCC value of 1.0. In order to best illustrate the essential points, stratified sampling is used, so that the IR of the train and test sets are identical (a validation set is not needed here due to the simplicity of the example).

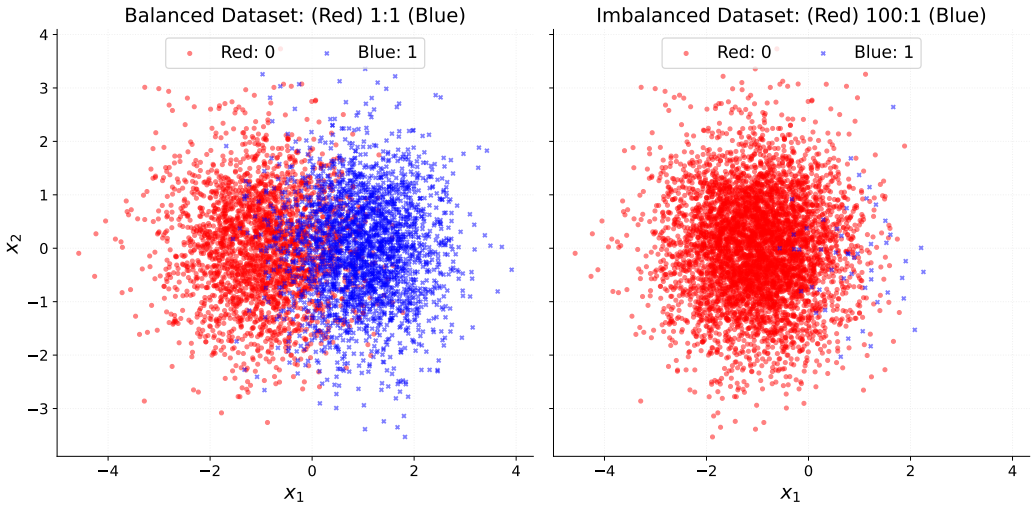

Figure 3.1: Two-dimensional balanced (left) and imbalanced (right) sample binary-classification datasets.

For each dataset, a logistic regression model is trained using the binary cross

entropy (BCE) loss objective, and train and test errors, together with key statistics derived from the confusion matrix (MCC and G-Mean for train and test set; Recall, Precision, and Specificity for test set) are plotted, as a function of training epoch, in Figure 3.2, which provides a side-by-side comparison of train and test performance.

It can be seen that for the balanced dataset, train and validation MCCs converge to a relatively high value of around 0.8, while for the imbalanced case the MCC is much lower, with a converged value of around 0.4. A similar observation can be made about the G-Mean behaviours. The lowest row of figures in Figure 3.2 provides an explanation for this: in the imbalanced case, Recall is much less than Precision, indicating an unwanted focus on the majority negative class, with a large proportion of the minority positive class being wrongly predicted.

This example has been sufficient to demonstrate that the difficulties connected to class imbalance manifest very quickly in classification problems, as the example was a simple two-dimensional one, in which - setting aside the overlap - the examples could have been separated easily by a linear model. However the behaviour of the BCE loss function does not make these difficulties clear; the loss in fact drops faster in the imbalanced case, as the model is solving a simpler - but erroneously constructed - problem, in which it is satisfactory to assign many minority examples to the majority class. The most obvious solution is to use a loss that in contrast *does* register clearly when a model is heading toward the trivial solution of assigning many, or all, minority examples to be majority class, such as loss function built from a classification measure such as the G-Mean or MCC, which would heavily penalise this kind of behaviour.

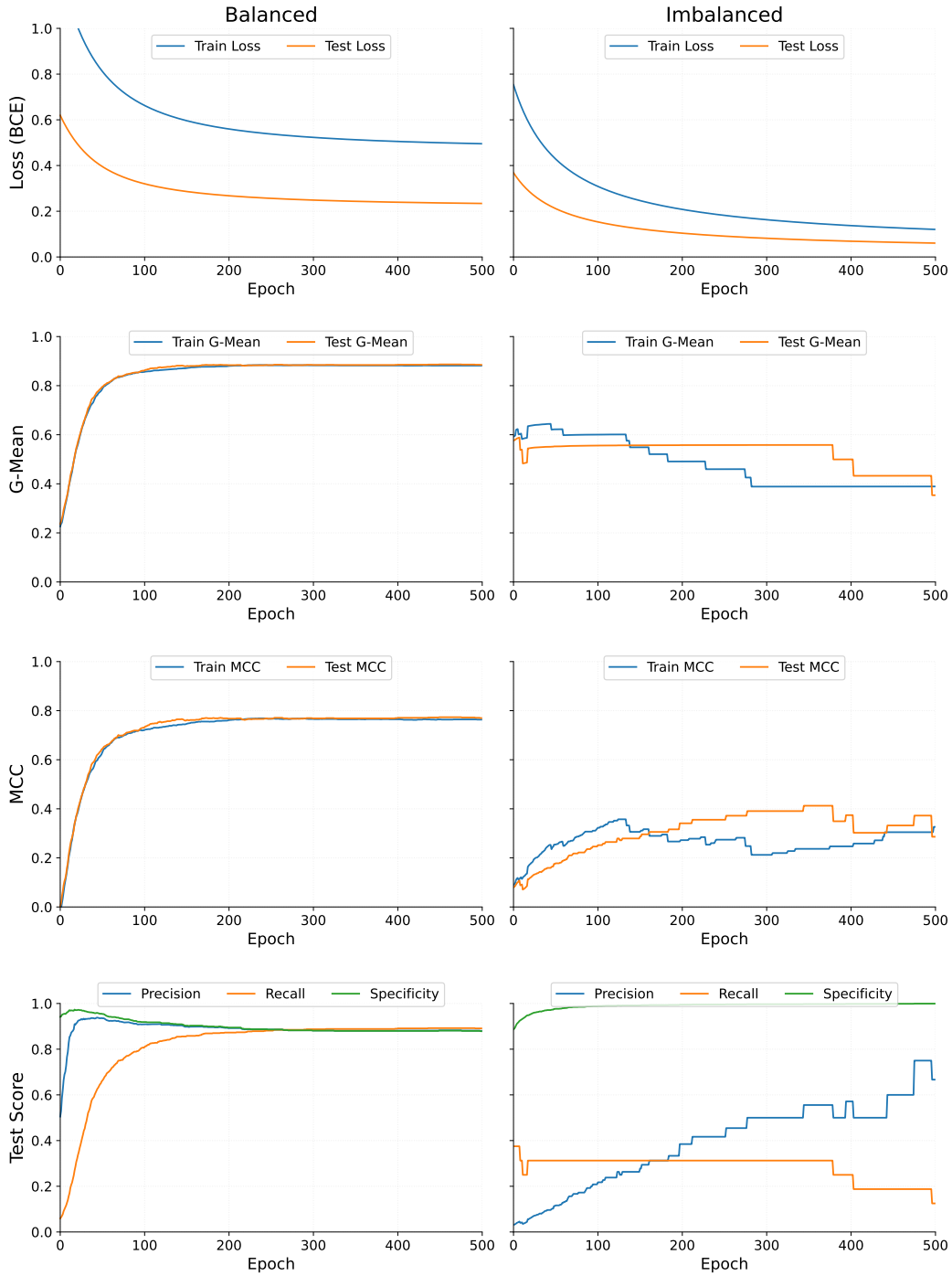

Figure 3.2: Train and test model performance as a function of training epoch for the balanced (left column) and imbalanced (right column) sample binary-classification datasets.

3.3 A New Training Metric for Classification

This section introduces the approximated G-Mean, the first of the novel loss functions proposed by this thesis, which will be used in the experiments of the current chapter.

3.3.1 The Approximated Confusion Matrix

As has already been explained, traditional measures derived from the confusion matrix are non-differentiable and thus not suitable for direct optimisation using gradient backpropagation. However, as was noted in the introduction to this chapter, if we replace the predicted class-labels of the model (which are used to construct the traditional confusion matrix) with their probabilistic equivalents, we obtain an 'approximated' confusion matrix which is in fact differentiable with respect to the predictions and model parameters. This approximated confusion matrix can be expressed as:

$$CM_{apx} = \begin{pmatrix} TN_{apx} & FP_{apx} \\ FN_{apx} & TP_{apx} \end{pmatrix} = \begin{pmatrix} \sum_i^n (1-\hat{y}_i)\cdot(1-y_i) & \sum_i^n \hat{y}_i \cdot (1-y_i) \\ \sum_i^n (1-\hat{y}_i)\cdot y_i & \sum_i^n \hat{y}_i \cdot y_i \end{pmatrix}, \tag{3.1}$$

in which $\hat{y}_i$ and y_i are the predicted and target values for the i^{th} example, and the sums are over the set of examples for which the approximated matrix is required. (The above form of the approximated confusion matrix assumes that outputs and targets are in the range $\in [0,1]$; other constructions are possible, for example using the range $\in [-1,1]$, and are a straightforward adaptation.)

From CM_{apx} one can generate a plethora of 'approximated' performance measures including all those based on a confusion matrix introduced in Chapter 2. This chapter, however, will focus on the approximated G-Mean, one of the simplest of these measures, and show how it can be effectively used as a loss function for a set

of benchmark problems, outperforming both the MSE and BCE losses.

3.3.2 The Approximated G-Mean Loss Function

The G-Mean, the simplest of the measures introduced in Chapter 2, and the choice for this chapter's experiments, tries to maximise the accuracy on each of the classes while keeping these accuracies balanced.

At first, the arithmetic mean (Balanced Accuracy) of each class-specific accuracy might have appeared an even simpler and better choice, but the G-mean is preferred here because it is a more pessimistic measure and handles extreme scenarios better. From the Arithmetic Mean-Geometric mean theory (AM - GM inequality), the balanced accuracy is always greater than or equal to the G-Mean:

$$\frac{x+y}{2} \geq \sqrt{x \times y} \tag{3.2}$$

$$\textit{Balanced-Accuracy} \geq \textit{G-Mean} \tag{3.3}$$

Hence when the specificity and sensitivity are equal, the Balanced Accuracy and the G-mean are equal. However, the measures diverge the bigger is the difference between sensitivity and specificity, due to the fact that the geometric mean is affected more by the lower value. The G-mean is also chosen since, unlike other possible choices such as the MCC and F-Measure, its approximated form has a relatively simple derivative. In practice, we choose in this chapter to optimise the *square* of the G-Mean, on the basis that this choice will have the same effect as optimising the G-Mean itself while resulting in a simpler derivative. The necessary expression is

$$J_{GMN^2} = 1 - \textit{G-Mean}^2_{apx} = 1 - \frac{\sum_i (1-\hat{y}_i)(1-y_i) \cdot \sum_i \hat{y}_i y_i}{m_0 \cdot m_1}, \tag{3.4}$$

in which m_0, m_1 are the numbers of negative, positive examples, respectively. The

simplicity of the denominator in this instance arises from the fact that the denominators of the expressions for Recall (Sensitivity) and Specificity simply count the numbers of positive and negative examples, respectively. The necessary derivatives with respect to network outputs can then be either derived algebraically or obtained via automatic differentiation, and are given below

$$\frac{\delta J_{GMN^2}}{\delta \hat{y}_i} = \frac{(1-y_i)\cdot\sum_j \hat{y}_j y_j - y_i \cdot \sum_j (1-\hat{y}_j)(1-y_j)}{m_0 \cdot m_1}. \tag{3.5}$$

This derivative can be used to calculate derivatives with respect to weights, and hence weight changes, via the chain rule of differentiation. We note that while the form of the derivative above is relatively simple, it does still imply a pre-computation of the $\hat{y_p}$ values in order to know the values of the summations:

$$\begin{aligned} TP_{apx} &= \sum_j \hat{y}_j \cdot y_j, \\ TN_{apx} &= \sum_j (1-\hat{y}_j)\cdot(1-y_j). \end{aligned} \tag{3.6}$$

(Note: In the original presentation of this work [73], a tanh activation function was used in the output layer of the neural networks, rather than the more usual sigmoid. The presentation here has been adapted to better match that of the loss functions used elsewhere in the thesis; transformation of equations 3.5, 3.4 and 3.6 to use tanh is straightforward.)

3.4 UCI Dataset Experiment

This section will demonstrate the effectiveness of the new loss function over a wide range of benchmark binary classification datasets from the UCI repository [92] which contain varying degrees of class imbalance.

3.4.1 Datasets

The 30 chosen datasets were downloaded from the UCI repository [92], with class imbalance ratios ranging from 5.55 (*dermatology-6*) to 129.44 (*abalone19*), feature dimensions ranging from 6 (e.g. *car-good*) to 41 (e.g. *kddcup-guess-passwd-vs-satan*), and numbers of examples ranging from 148 (*lymphography-normal-fibrosis*) to 2935 (*abalone19*). A more detailed summary of each dataset is provided in Table 3.1.

3.4.2 Experiment Design

The experimental design was kept as simple as possible, in order to focus on the key question of whether a confusion matrix based loss function would be more effective, for higher IRs, than a traditional one; it was not the intention of this work to solve each problem maximally well, per se.

Prior to training, the following procedure was conducted on each dataset:

1. **Initial train-test k-fold split**. Each dataset was first shuffled and then divided into 5 cross folds, each allocating 20% of the examples to testing.

2. **Training-validation split**. For each fold, the remaining 80% of the data was subdivided into 75% training and 25% validation, using stratified sampling to ensure similar class distributions for training and validation.[1] (Note that the separations for constructing the test sets were not stratified).

3. **Standardisation**. All inputs were standardised by subtracting the mean and dividing by the standard deviation for that input (feature) in the training set.

For each fold (independent test experiment), three networks were trained: one using the G-Mean loss, one using the standard binary cross entropy (BCE) loss, and

[1]The number of chosen folds was arbitrary, but the size of the validation set was selected so that the validation and test sets were of the same size.

Dataset Name	#Features	#Examples	Class Imbalance
abalone-17_vs_7-8-9-10	8	2338	39.31
abalone19	8	4174	129.44
abalone-19_vs_10-11-12-13	8	1622	49.69
abalone-20_vs_8-9-10	8	1916	72.69
abalone-3_vs_11	8	502	32.50
car-good	6	1728	24.04
car-vgood	6	1728	25.58
cleveland-0_vs_4	13	177	12.60
dermatology-6	34	366	5.55
ecoli3	7	336	8.60
flare-F	11	1066	23.79
glass2	7	214	11.60
glass6	9	214	6.40
kddcup-guess_passwd_vs_satan	41	1642	29.98
kr-vs-k-one_vs_fifteen	6	2244	27.77
kr-vs-k-three_vs_eleven	6	2935	35.23
kr-vs-k-zero_vs_eight	6	1460	53.07
kr-vs-k-zero_vs_fifteen	6	2244	80.22
kr-vs-k-zero-one_vs_draw	6	2901	26.63
lymphography-normal-fibrosis	18	148	23.67
segment0	19	2308	6.00
shuttle-2_vs_5	9	3316	66.67
shuttle-6_vs_2-3	9	230	22.00
vowel0	13	988	10.00
winequality-red-4	11	1599	29.17
winequality-red-8_vs_6	11	656	35.44
winequality-red-8_vs_6-7	11	855	46.50
winequality-white-3-9_vs_5	11	1482	58.28
yeast5	8	1484	32.73
yeast6	8	1484	41.40

Table 3.1: Summary of the 30 chosen UCI datasets used for the G-Mean experiment.

one using the mean squared error (MSE) loss. Both the hidden and output activations were set as tanh and the output binarised with a decision threshold of 0 (since the *tanh* produces an output in the range [-1,1]). The architecture of the network was

constrained to a single hidden layer with the number of nodes equal to the average of the number of inputs and output dimensions.

The weights of the network were trained using a bold-driver backpropagation technique similar to that of [93]. For each epoch, this algorithm attempts to adjust the weights in the direction minimising the error. Within an epoch the learning rate can be reduced if the initially attempted step resulted in an increase in error. After a successful reduction in error, the learning rate is conversely increased. The training process was run for a fixed 1,000 epochs but at each epoch the error was evaluated on the validation set and weights of the best performing epoch were stored as the final weights. After training, the model is fed the test examples and its out-of-sample classification performance evaluated using the Matthews Correlation Coefficient (MCC, equation 2.11), the MCC being chosen as an independent (from the G-Mean-based loss function) measure that like the G-Mean would penalise over-assignment to the majority class.

MCCs for the competitor methods were compared to those for G-Mean-based training using their *relative underperformance to the GMN-trained net*, calculated as:

$$\xi_{ALT} = \frac{MCC_{ALT} - MCC_{G\text{-}Mean}}{|MCC_{G\text{-}Mean}|} \tag{3.7}$$

where the subscript refers to the network's cost function and $ALT \in \{RMSE, BCE\}$. Underperformance was chosen instead of outperformance in order to avoid divide-by-zero errors, as for several datasets the ALT cost functions gave zero MCC scores as all instances were assigned to the same class. In our observation the working range of ξ_{ALT} was from -1 (ALT score zero, GMN score > 0) to slightly above zero (ALT score slightly exceeding GMN score). An underperformance value of -1 will represent the strongest evidence in our test set for the superiority of GMN.

3.4.3 Results

Figure 3.3 shows the average competitor underperformances ξ_{ALT} across five different (dataset shuffling and weight) initialisations as a function of IR. (An MCC score for a run is for these experiments based on the combined confusion matrix of all five folds, not on an average of each fold). For IR < 20 all three cost functions behave similarly, but as the IR becomes more extreme there are more examples of the clear superiority of the G-Mean. In the most extreme example (*abalone19*), with an IR of 129.44, both the MSE and BCE trained networks make no attempt to predict the minority class and thus achieve zero MCC scores, while in contrast the G-Mean attains a positive MCC of 0.09 ± 0.04.

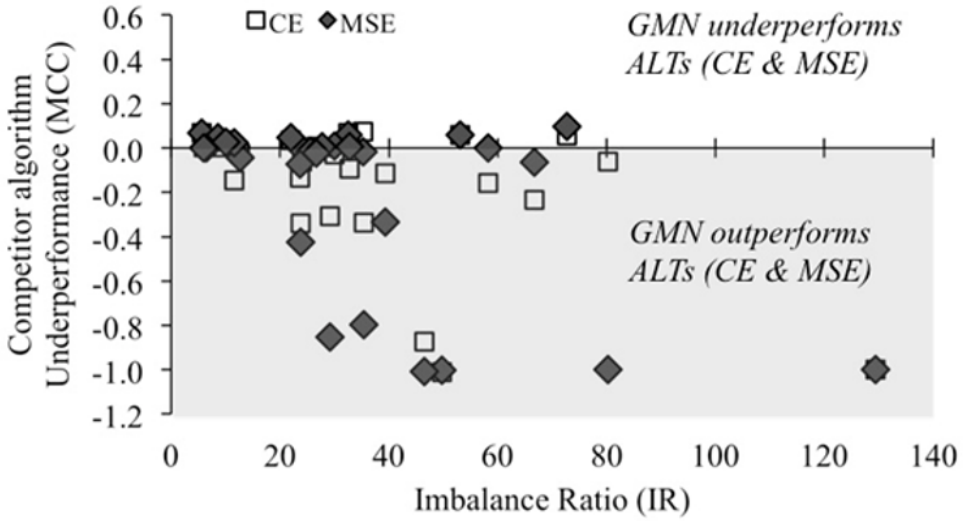

Figure 3.3: Underperformance (measured by MCC) of both MSE and BCE relative to GMN, plotted against imbalance ratio (IR). GMN outperforms the competitor cost functions everywhere in the shaded region

Though increasing IR generally leads to better relative performance of the G-Mean loss, there are exceptions. One such is *abalone-20-vs-8-9-10* (IR=72.69), for which both ALTs marginally outperform the G-Mean. This case was investigated with an additional 55 runs, discovering that even in such an apparently unfavourable case there may be benefits to the use of the G-Mean loss, as evidenced in Figure 3.3.

It can be seen there that while the two ALT MCC scores have a higher average, there is much more variance, and more instances for which both ALTs have an MCC of zero (all cases assigned to the majority class), and thus no discretionary power.

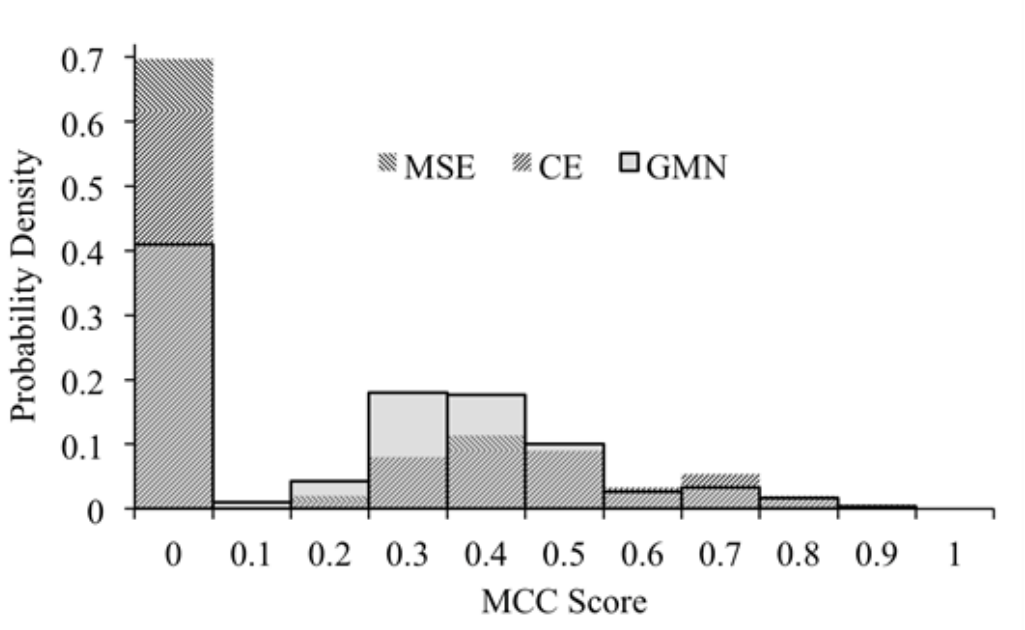

Figure 3.4: Distribution of out-of-sample scores across all random seeds/k-folds (300 total) for GMN, BCE, and MSE cost functions, for the exceptional dataset *abalone-20-vs-8-9-10*.

In summary, the advantage of the G-Mean loss over the conventional binary cross entropy (BCE) and MSE losses has been demonstrated on 30 imbalanced UCI datasets (IRs from 5.55 to 129.44). For IRs less than 20, the three cost functions performed similarly, but higher imbalances showed a clear benefit of using GMN. The GMN scores were also found to be less sensitive to initial conditions

3.5 Comparison to Data-Level Approaches: Pump & Dump

The results of Section 3.4 were encouraging regarding the use of an approximated G-Mean-based loss to address the challenge of class imbalance. However, changing

the loss function is just one of several possible approaches; as mentioned in Section 2, data-level re-sampling approaches such as undersampling and SMOTE have become extremely popular tools for imbalanced learning given their documented successes, and that they are generally very simple procedures to implement. Therefore, it is useful to compare the performance of the G-Mean loss (and standard loss functions) against the most common data-level approaches. For this experiments of this section, a more complex real-world is selected for benchmarking, connected to the problem of 'pump and dump' events in finance.

3.5.1 Pump & Dump

A 'pump and dump' scheme is a type of financial trading exploitation which involves an entity accumulating a large amount of a target currency, stock or, in this case, cryptocurrency (referred to here also as a crypto-asset, or simply crypto), with a low market capitalisation and a low unit price, and subsequently promoting its purchase as an opportunity for substantial future returns. (In some cases of pump and dumps, promotions may claim to have inside information of future events that are expected to cause price increases.) The promotion is designed to create a significant buying demand which increases the price (i.e. the 'pump'). The promotion continues, and while others are still trying to enter, the initial position is fully unwound for a considerable profit. The promotion is then halted, and the price then crashes (i.e. the 'dump'), often incurring significant losses for those slow to react.

Within cryptocurrency markets, a variation of the above has become popular. Groups are formed on social chat platforms where low market capitalisation coins are selected and advertised by administrators. Members of the group then engage in intensive purchasing within a short time span (within a couple of minutes). The group's mutual buying pressure causes a sudden rise in the price of the asset. Unlike in more traditional markets, where those purchasing are doing so on false information

(being duped by the promoter), members in these cryptocurrency groups are aware of what is occurring and are willing to participate in the belief they can sell to others who have been slower to act on the information.

Numerous cryptocurrency pump and dump groups have been established with member counts ranging into the tens of thousands. The groups are advertised widely within the crypto social media ecosystem - across Twitter, Slack, Discord and Reddit – to attract new members. Most pump and dump groups follow a similar process: the coordinating entity sends out one (or many) messages advertising the time and trading exchange upon which the "pump" will occur, but not which coin is being chosen. At the specified time, the coordinating entity informs the group of which target cryptocurrency to buy. People are willing to complete a purchase (even at a premium compared to a few minutes before) based on their belief they can sell at a higher price while the coin is being pumped. If members do make money, it is likely that they have done so at the expense of someone else in the group who has been slower at buying into the cryptocurrency. Some groups have tried to address this by prolonging price increases. One mechanism to do this is encouraging their members to promote the chosen cryptocurrency on social media platforms after it has been announced.

3.5.2 Dataset

A pump and dump raw dataset, constructed from historical cryptocurrency pump and dump events, was generously provided by Jiahua Xu & Benjamin Livshits, the authors of [94]. In this paper, Xu & Livshits present a case study of a group of pump and dump events that occurred within cryptocurrency markets between 2018-19. Telegram message history from over 300 channels from June 17th 2018 to February 26th 2019 was collected (via the Telegram API) and used to identify 412 announced pump events. Telegram is the most common mobile chat platform

that pump and dump groups are formed on; due to its high level of anonymity and encryption, perpetrators are difficult to identify. The format of these Telegram messages announcing a pump would include both the exchange venue and scheduled time but not the target coin itself (for example, 'Binance @ 11CEST'). As such, the target coin could be one of any traded on the target venue. The chosen coin is instead announced immediately before the event.

The authors of [94] frame a classification problem as follows: for a specific pump event, they group the available tokens actively trading on the target exchange and assign all of them a negative label (0), except for the target token which is given a positive label (1). (Of course, the identity of this target token is only known after the event.) Features of the coins and their recent market activity are constructed. Repeating this process across all pump events, a dataset can then be constructed which can be used to train a classifier. In [94] this dataset is used to demonstrate how an ensemble decision tree classifier could be trained to predict these events and, consequently, form a trading strategy. However, since the problem is highly-class-imbalanced, which created difficulties for learning, a profitable strategy required a post-hoc threshold adjustment of the classifier output. The dataset is thus of clear interest as a proving ground for the G-Mean loss function, which should be better able to handle the class imbalance.

The dataset used here was partitioned in a manner similar to that of the original authors, but it was decided to combine the authors' train (Jul 21 - Oct 10) and validation (Oct 10 - Oct 29) periods, leaving just one train dataset (which was, however, during hyperparameter optimisation divided into five folds), and one test dataset. Detailed information about the features, which are kept the same as in the cited paper, can be found in [94]. Both training and test features are pre-processed by removing the median and dividing the inter-quartile range (IQR) exhibited in the

training set. Table 3.2 provides a summary of the final training and test dataset; the natural class imbalance is extremely high at 500+ for both subsets.

Table 3.2: Pump & Dump dataset train and test partitions

	#Total	#Features	#No Pump	#Pump	Class Imbalance
Train	37,819	54	37,759	60	630.31
Test	9,747	54	9,728	19	513.00

3.5.3 Experiment Design

Given the high level of class imbalance present in the dataset, it is theorised that the G-Mean may be a better loss function to use than other, standard, functions. However, the expanded aim of this section is to examine whether a change of loss function to the G-Mean can produce competitive results when compared to data-level approaches. The experiment is therefore split into three phases:

- **Baseline Experiment**. The first setup is designed to evaluate the performance of networks trained using the G-Mean loss function against the two most commonly used alternatives: Binary Cross Entropy (BCE) and Mean Squared Error (MSE) - as in the UCI work - on the original, unadjusted dataset.

- **Comparison with Undersampling**. The second setup explores the use of undersampling, a data-level strategy to reduce the level of class imbalance. The training set is progressively undersampled and the results for the resulting models compared on the original test set.

- **Comparison with SMOTE**. The final setup uses a SMOTE resampling technique, (first introduced in Chapter 2), to create new synthetic training samples of the minority class. Again, the results, this time for a single model, are evaluated with respect to the original, unmanipulated, test set.

3.5.4 Results

3.5.4.1 Part I: Baseline Experiment

The first phase involves no competitor data-level approaches and is intended to serve as a baseline, comparing the performance of networks trained using the G-Mean loss function against two commonly used alternatives, the Binary Cross Entropy loss (BCE) and Mean Squared Loss (MSE). Using the train-test partition described previously, three single hidden-layer neural networks are trained; one for each candidate loss function.

To establish the best stopping epoch for training, the training set is first partitioned into five folds and the stopping epoch taken to be the average of the best epoch (in terms of MCC validation score) across each fold. The final model is trained on the entire training dataset for the number of epochs determined using the k-fold splits. The experiment is repeated 25 times with different weight initialisations and the out-of-sample test results for each model is shown in Table 3.3.

Table 3.3: Test classification results for the original pump and dump dataset. Performance is averaged over 25 runs, with standard deviations shown in brackets.

	BCE Training	**MSE Training**	**G-Mean Training**
Accuracy	0.998 (0.001)	0.952 (0.078)	0.999 (0.000)
Precision	0.594 (0.486)	0.018 (0.046)	0.673 (0.026)
Recall	0.095 (0.146)	0.184 (0.243)	0.474 (0.000)
MCC	0.181 (0.137)	0.040 (0.080)	0.564 (0.011)

If you were to examine the results on out-of-sample accuracy alone, all three candidates appear to perform very well, with $> 95\%$ accuracy. However, there is a big difference in the Recall and MCC scores, all of which consider how errors are distributed across classes, rather than the overall number of errors: GMN Training very clearly outperforms BCE and MSE for these measures, and this out-performance

is statistically significant using a one-sided t-test, with a p-value < 0.01 when comparing GMN training to both BCE and MSE training. The null hypothesis for the one-sided t-test in this context would be that there is no significant difference in the performance (in this case in terms of Recall and MCC scores) between the GMN training and the BCE/MSE training. In other words, any difference observed in the sample data could be due to chance alone and not a true difference in the populations being compared.

In relation to Precision, GMN training again statistically significantly (using the same test) outperforms MSE training. However, it does not outperform BCE training. This is because BCE training is very effective here on the majority class, though it achieves its accuracy on the majority at the expense of misclassifying minority examples, as can be clearly seen, for example, in the quoted MCC values.

3.5.4.2 Part II: Comparison with Undersampling

The outperformance of models trained with the GMN loss versus standard loss functions on the pump and pump dataset fits with the premise that the GMN is well-suited to highly class-imbalanced datasets. However, instead of using the GMN, a data-resampling technique could instead be employed. One such obvious example is undersampling.

To compare the GMN against undersampling, the training data is progressively undersampled, creating new training datasets, each with a lower degree of class imbalance, but smaller in size. Independent neural networks are fitted on each of these undersampled training datasets for each of the two candidate competitor loss functions (MSE and BCE), and their results compared to GMN training on the original test set. Starting from an undersample ratio of 0.002 (corresponding to a near-original level of 500:1 class imbalance) and increasing this to 1, a total of 28 distinct synthetic datasets are produced, which are summarised in Table 3.4.

The goal is to determine if the superior performance of the GMN loss function, demonstrated in the baseline experiment, declines as the class imbalance in the training data becomes less pronounced.

Undersample Ratio	CI	#Non-Pump	#Pump	#Total
0.00	500.00	30,000	60	30,060
0.02	45.45	2,727	60	2,787
0.04	23.80	1,428	60	1,488
0.06	16.12	967	60	1,027
0.08	12.18	731	60	791
0.10	9.80	588	60	648
0.12	8.18	491	60	551
0.14	7.03	422	60	482
0.16	6.17	370	60	430
0.18	5.48	329	60	389
0.20	4.95	297	60	357
0.25	3.97	238	60	298
0.30	3.30	198	60	258
0.35	2.83	170	60	230
0.40	2.48	149	60	209
0.45	2.20	132	60	192
0.50	1.98	119	60	179
0.55	1.80	108	60	168
0.60	1.65	99	60	159
0.65	1.53	92	60	152
0.70	1.42	85	60	145
0.75	1.32	79	60	139
0.80	1.23	74	60	134
0.85	1.17	70	60	130
0.90	1.10	66	60	126
0.95	1.05	63	60	123
1.00	1.00	60	60	120

Table 3.4: Pump & Dump datasets resulting from majority-class (non-pump) under-sampling.

Figure 3.5 plots the under-sample ratio against the Matthews Correlation Co-

efficient on the test set, averaged over 25 runs. It is very clear from this figure that in relation to the MCC value, widely considered a 'gold standard' for measuring performance under class imbalance, that GMN training is superior to using a standard loss function and random-undersampling the training set, as this can lead to loss of useful information, as is evidently the case here for any but a very small level of majority class undersampling.

While a small amount of undersampling, from Figure 3.5, is clearly of benefit to BCE and MSE training, and could also have benefited GMN training, this was not used with the GMN as the purpose of using such a novel loss function was specifically to avoid the need for manipulation of the data.

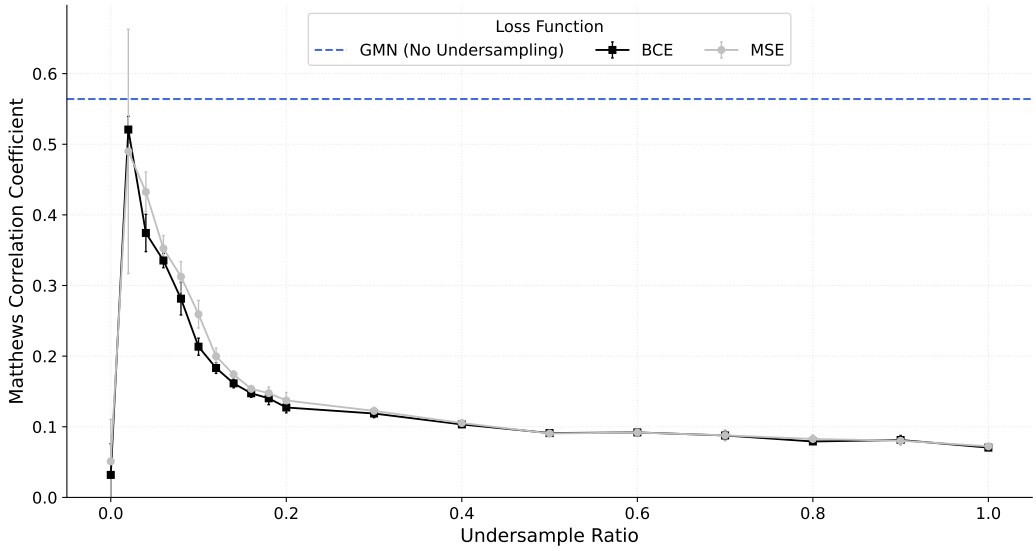

Figure 3.5: Test MCC scores on pump and dump data for models trained with BCE and MSE losses, for progressively undersampled train sets, and for GMN loss without undersampling (dotted line).

3.5.4.3 Part III: Comparison with SMOTE

The final experiment is to see whether the GMN loss function can compete with more complex data-level approaches for dealing with class imbalance; in this case the Synthetic-Minority Oversampling (SMOTE) algorithm, which was discussed in

Chapter 2. For the MSE and BCE loss functions, SMOTE is applied to the training dataset while keeping the test set unaltered, and these results are compared for the test those obtained from GMN training with an unaltered train set.

In order to assess whether the results are statistically significant, we formulate a hypothesis test framework to evaluate the performance differences. The results are from different random initialisations on the same testing set, and a one-sided t-test is used to determine if a given metric for GMN training is, on average, statistically significantly superior to that for a competitor.

Such testing evidences that GMN is statistically significantly better than both MSE+SMOTE and BCE+SMOTE in relation to the key measure of MCC and in relation, also, to Precision (with p-values ¡ 0.01). GMN does not outperform its SMOTE-augmented competitors in relation to Recall (p ¿ 0.05). However, the practical consequences of the quoted results need to be considered: the superiority of the SMOTE-augmented competitors in relation to Recall would be of value only if there was a very low penalty associated with acting on an incorrect model prediction (i.e. low false positive error cost), as the Precision of such predictions are below 1.5% for both MSE+SMOTE and BCE+SMOTE; given the high cost of executing the necessary trading strategy this would in fact represent an unacceptable level of risk. Overall, it can be concluded that GMN training is superior to SMOTE-augmented standard training.

3.6 Discussion

GMN training has in this chapter shown itself to be a highly competitive training algorithm when compared to training based on standard loss functions, both for a wide range of UCI benchmark datasets, and for data relating to a certain kind of cryptocurrency market manipulation, referred to as a ‘pump and dump’. In the case

	BCE (+ *SMOTE*)	**MSE** (+ *SMOTE*)	**GMN**
Accuracy	0.898 (0.023)	0.879 (0.032)	0.999 (0.000)
Precision	0.013 (0.001)	0.012 (0.002)	0.673 (0.026)
Recall	0.695 (0.065)	0.726 (0.092)	0.474 (0.000)
MCC	0.087 (0.004)	0.083 (0.003)	0.564 (0.011)

Table 3.5: Test classification results for the SMOTE versus GMN pump and dump dataset experiment. Performance is averaged over 25 runs, with standard deviations shown in brackets.

of this latter dataset, with a challenging IR > 500, the competitive advantage of the GMN loss was evident even when the standard loss functions were augmented by data-level interventions (random undersampling and SMOTE).

Moving forward from the promising results of this chapter, two avenues of work would immediately suggest themselves. Firstly, given that GMN training has here shown its largest benefits in cases of high IR (the highest-IR datasets in the UCI collection, plus the even higher IR pump and dump dataset), it would of clear interest to consider datasets with still higher imbalance ratios; results using the GMN loss function on datasets with IRs up to 4,000 can be seen in Chapter 5. Secondly, the extension of the GMN from binary classification problems to multi-class problem could be examined, especially as [95] have shown that existing approaches, such as data resampling, formerly believed to be effective for addressing the multi-class class imbalance problem, may in fact only be effective for two-class datasets. In addition to introducing new classification-focused loss functions, namely the F1-Score and MCC, Chapter 4 will also focus on extending the GMN loss function, introduced in this chapter, to accommodate multi-class scenarios.

Chapter 4

Classification-Focused Training Metrics: Alternatives & Extension to Multi-Class Scenarios

This chapter extends the work of the previous chapter in two ways. First, it explores alternative classification-based loss functions to the approximated Geometric Mean (GMN), introduced in Chapter 3. The G-Mean loss performed well under situations of high class-imbalance; however, it is possible that loss functions based on other valuable measures of imbalanced performance, such as the Matthews Correlation Coefficient (MCC), might perform even better. The performance of these alternatives will be assessed here on two public medical image segmentation datasets. These are larger and more complex datasets than those considered in the previous chapter, and chosen here as good candidates because of their relatively high pixel-level class imbalance. Notably, these experiments implement two other loss functions which have received recent attention in deep learning and computer vision literature for their out-performance on class-imbalanced datasets, the Dice Loss and Focal Loss, with results demonstrating that the approximated confusion matrix based losses

outperform even these current state-of-the-art losses. Second, this chapter broadens the application of novel, confusion matrix-based methods to multi-class situations, utilizing them for long-tailed recognition problems in the image domain. Unlike the medical image segmentation problems, these have labels assigned at the image level rather than the pixel level. The results demonstrate improved performance over the current state of the art, with the superior performance of the new loss functions becoming more pronounced as class imbalances increase.

4.1 Alternatives to the Geometric Mean

In Chapter 3, Equation 3.1 introduced the concept of an approximated confusion matrix, a differentiable variant of the normal confusion matrix which replaces predicted class-labels with the underlying model's probabilistic outputs. It was shown that the components of this approximated confusion matrix, (TN_{apx}, FP_{apx}, FN_{apx} and TP_{apx}) could be used both to create performance metrics that give more precise information about the progress of training, and also, being now differentiable, be used as a *classification-focused* loss training objective in the optimisation of a neural network model. The performance measure chosen in the previous chapter as the basis for a novel loss function was the G-Mean (Equation 2.10), with the resulting GMN loss function (Equation 3.4) being shown empirically to produce good results on many datasets of varying class imbalance. The G-Mean was initially chosen for two reasons. Firstly, because it is a measure which tries to maximise the accuracy on each class while keeping these accuracies balanced, and hence is regarded as a good measure for assessing performance under class imbalance. Secondly, because its (approximated) derivative form reduces to something relatively simple and thus might lead to a better gradient loss surface, less prone to getting stuck in local minima than other, more complicated, measures. However, it was noted in

the previous chapter that it is in fact possible to construct many other loss functions, based on a plethora of other confusion-matrix based performance measures, and it is thus at least possible that some of these might outperform the G-Mean loss, rendering them worthy of the investigation which is the topic of this section. These comparison experiments will focus on two medical image segmentation problems, with data, methods, and results to follow in later subsections.

4.1.1 The F1 Score and MCC Loss Functions

Two new loss functions will be considered, based on the F1-score and the Matthews Correlation Coefficient (MCC), the former due to its popularity and common recommendation for performance assessment under class imbalance, and the latter because, particularly in recent years, it has received recognition as the 'gold standard' for model performance and comparison [47].

From Equation 2.7, the *approximated* F1-score can be expressed as

$$\begin{aligned} \textit{F1-Score}_{apx} &= \frac{TP_{apx}}{TP_{apx} + \frac{1}{2}(FP_{apx} + FN_{apx})} \\ &= \frac{2TP_{apx}}{2TP_{apx} + FP_{apx} + FN_{apx}} \\ &= \frac{2\sum_i \hat{y}_i \cdot y_i}{2\sum_i \hat{y}_i \cdot y_i + \sum_i \hat{y}_i \cdot (1 - y_i) + \sum_i (1 - \hat{y}_i) \cdot y_i}, \end{aligned} \tag{4.1}$$

and we can thus easily create a new F1-Score loss function,

$$J_{F1} = 1 - \textit{F1-Score}_{apx}, \tag{4.2}$$

where the derivatives with respect to network outputs can, in a similar manner to the G-Mean approximated loss, be obtained via automatic differentiation or derived analytically:

$$\begin{aligned}
&\frac{\partial J_{F1}}{\partial \hat{y}_i} \\
&= -\frac{[2\sum_j \hat{y}_j \cdot y_j + \sum_j \hat{y}_j(1-y_j) + \sum_j (1-\hat{y}_j)y_j]2y_i - [2\sum_j \hat{y}_j \cdot y_j][2y_i + (1-y_i) - y_i]}{[2\sum_j \hat{y}_j \cdot y_j + \sum_j \hat{y}_j(1-y_j) + \sum_j (1-\hat{y}_j)y_j]^2} \\
&= -\frac{[2\sum_j \hat{y}_j \cdot y_j + \sum_j \hat{y}_j(1-y_j) + \sum_j (1-\hat{y}_j)y_j]2y_i - 2\sum_j \hat{y}_j \cdot y_j}{[2\sum_j \hat{y}_j \cdot y_j + \sum_j \hat{y}_j(1-y_j) + \sum_j (1-\hat{y}_j)y_j]^2}.
\end{aligned} \tag{4.3}$$

Similarly, the *approximated* Matthews Correlation Coefficient (MCC) can be be derived to be

$$\begin{aligned}
MCC_{apx} &= \frac{TP_{apx} \times TN_{apx} - FP_{apx} \times FN_{apx}}{\sqrt{(TP_{apx} + FP_{apx})(TP_{apx} + FN_{apx})(TN_{apx} + FP_{apx})(TN_{apx} + FN_{apx})}} \\
&= \frac{TP_{apx} \times TN_{apx} - FP_{apx} \times FN_{apx}}{\sqrt{m_0 \cdot m_1}[(TP_{apx} + FP_{apx})(TN_{apx} + FN_{apx})]^{1/2}}
\end{aligned}, \tag{4.4}$$

in which m_0, m_1 are the numbers of negative, positive examples, respectively. Hence the corresponding novel loss function can be derived as:

$$J_{MCC} = 1 - MCC_{apx}. \tag{4.5}$$

Again, the derivatives with respect to model outputs can be obtained either via automatic differentiation or derived analytically:

$$\begin{aligned}
\frac{\partial J_{MCC}}{\partial \hat{y}_p} = -\Bigg[&\frac{-(1-y)\cdot m_1 + y \cdot m_0}{(m_0 \cdot m_1 \cdot \sum_j \hat{y}_j)^{1/2}} \\
&-\frac{1}{2}\left(\frac{\sum_i (1-\hat{y}_i)(1-y_i) \cdot \sum_i \hat{y}_i y_i - \sum_i (1-\hat{y}_i) y_i \cdot \sum_j \hat{y}_j (1-y_j)}{\sqrt{m_0 \cdot m_1}(\sum_j \hat{y}_j \cdot \sum_j (1-\hat{y}_j)^{3/2}} \right. \\
&\left. \times [(m_0 + m_1) - 2\sum_j \hat{y}_j] \right) \Bigg]
\end{aligned} \tag{4.6}$$

4.1.2 Image Segmentation

Image segmentation is a technique where a pixel-wise mask is formed around an object contained in a given image, and represents an extension of image classification [96]. Put simply, it is the task of assigning labels to pixels. All pixels belonging to the same category have a common label assigned to them, which is referred to as the mask. There are different types of image segmentation, with the type relevant to the experiments carried out here being semantic segmentation (specifically, binary semantic segmentation). Semantic segmentation refers to the classification of pixels in an image into semantic classes (for example, in Figure 4.1b, belonging to a human figure, to sky, to sea, or to land) with no other information or context taken into consideration.

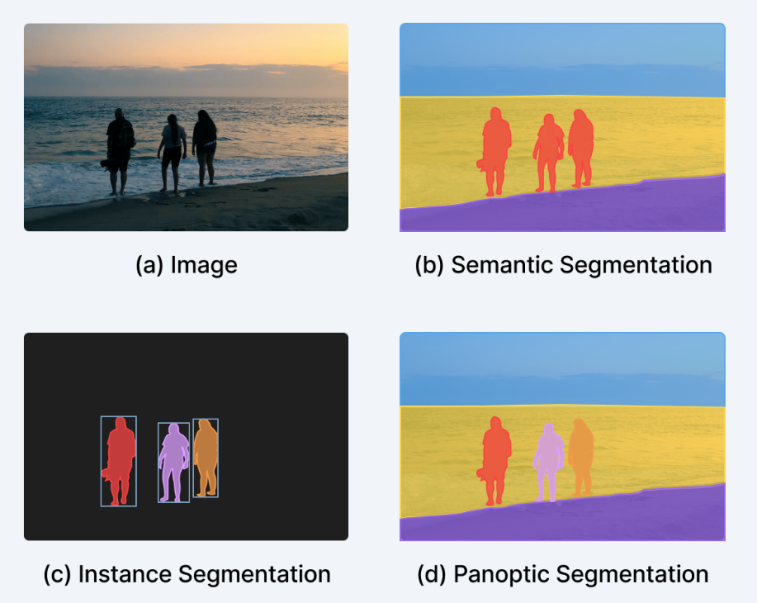

Figure 4.1: Types of image segmentation task. The datasets used here are examples of a binary form of semantic segmentation (b). Images reproduced from [3]

There are other forms of image segmentation, which are not explored in this work, but which are illustrated in Figure 4.1, such as instance segmentation, which instead classifies pixels into categories on the basis of *instances* rather than classes (in Figure 4.1c, belonging to one of a number - an instance - of a set of human

figures), or panoptic segmentation, which combines both semantic and instance segmentation so that each object in an image is segregated and identified.

The experiments here focus specifically on binary semantic segmentation within medical imaging. The confusion matrix based loss functions presented in the previous section will be compared to standard binary cross entropy and two of the most popular loss functions used for image segmentation, which will be described below.

The **Dice-Sørensen coefficient (DSC)** [97] is a statistic which gauges the similarities between two samples. It has become one of the most popular measures of region overlap in medical image analysis [98]. Given two datasets, X and Y, it can be defined as

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}, \tag{4.7}$$

where $|X|$ and $|Y|$ are the cardinalities of the two sets. However, in a binary classification setting, it can be re-written as

$$DSC = \frac{2TP}{2TP + FP + FN}. \tag{4.8}$$

Milletari *et al.* [99] were the first to use the Dice coefficient as a loss objective, defining it as the maximisation of

$$J_{Dice} = \frac{2\sum_i \hat{y}_i \cdot y_i}{\sum_i \hat{y}_i{}^2 + \sum_i y_i{}^2}, \tag{4.9}$$

and demonstrating that CNNs trained on imbalanced 3D image segmentation datasets using this loss function achieved good performance, requiring only a fraction of the processing time needed by other previous methods. By effectively reducing the number of measurements to the number of labels, the Dice loss has a lesser sensitivity to class imbalance. However, some authors, including [100], argue the Dice Loss

is not well suited to practical image segmentation, particularly in a medical setting, because it weights all regions of misplaced delineation equally and independently of their distance from the surface. For example, two inaccurate segmentations could have a similar DSC score if one were to deviate from the correct surface boundary by a small amount in many places while the other had a large deviation at a single point. Correcting the former would likely take a considerable amount of time as it would require redrawing almost all of the boundary while the latter could be corrected much faster, potentially with a single edit action.

The **Focal Loss (FL)** [75] is another type of loss objective that was introduced as a means of combatting extremely imbalanced datasets where positive cases were relatively rare. It can be seen as a dynamically weighted extension of the cross entropy loss,

$$J_{Focal} = -(1-p_t)^{\gamma}\log(p_t), \tag{4.10}$$

where p_t is the estimated probability for the ground truth class $p_t = \hat{y}\cdot y + (1-\hat{y})\cdot y$.

Since a lower p_s implies a larger error, Equation 4.10 gives more attention to harder examples. When $\gamma = 0$, the focal loss is actually the same as the vanilla cross entropy loss (BCE). One drawback is that γ needs to be pre-specified or tuned; following Lin et al. [75], a fixed $\gamma = 2$ is chosen for all the experiments of this subsection, as they find this value to be a good trade-off between hard and easy samples.

4.1.3 IRCAD & HuBMAP Datasets

Two public medical image segmentation datasets were chosen. The IRCADb01 (IRCAD) dataset [101] is composed of 3D CT scans containing hepatic tumours in 75% of cases; however, it should be emphasised that the problem here is to identify

pixels belonging to the tumour class, so that even when a tumour is somewhere present, most of the pixels will be non-tumour labelled. Tumour pixels account for just 6.71% of total pixels, creating an imbalance ratio of 14.897. The HuBMAP kidney image segmentation dataset is obtained from the Human BioMolecular Atlas Program (HuBMAP) [102], which is a project for the development of a framework to map the human body at single-cell resolution. It contains high-resolution images of PAS-stained kidney images with accompanying pixel-level masks of glomeruli. There can be over 600,000 glomeruli in each human kidney, each with a size range from 100-350um. The task is train a pixel-wise image segmentation model that can identify glomeruli regions of the images. Overall the dataset has a pixel-level imbalance ratio (IR) of 31.65 between glomeruli (minority) and non-glomeruli (majority) pixel masks.

Since both datasets are comprised of 20 very large image files (recall again that these experiments are being performed at the pixel level, and that these 20 files therefore represent a very large amount of data), a common image preprocessing technique known as tiling is used to cut up the images into smaller size images of a uniform size that can be better handled in memory. For both the IRCAD and HuBMAP datasets, tiles of size of 256×256 pixels are constructed, on the dual basis that the tile dimensions need to be divisible by 32 and that 256×256 is a common choice for work in this area. (Optimising the tile size is a research topic in its own right and so is beyond the scope of this experiment.) Also note that, in the case of the IRCAD dataset, the images are three dimensional and masks are first converted to two dimensional slices, by performing axial cuts along the z-axis, before tiling. Figure 4.2 provides an example of a single tile and its corresponding mask for each dataset.

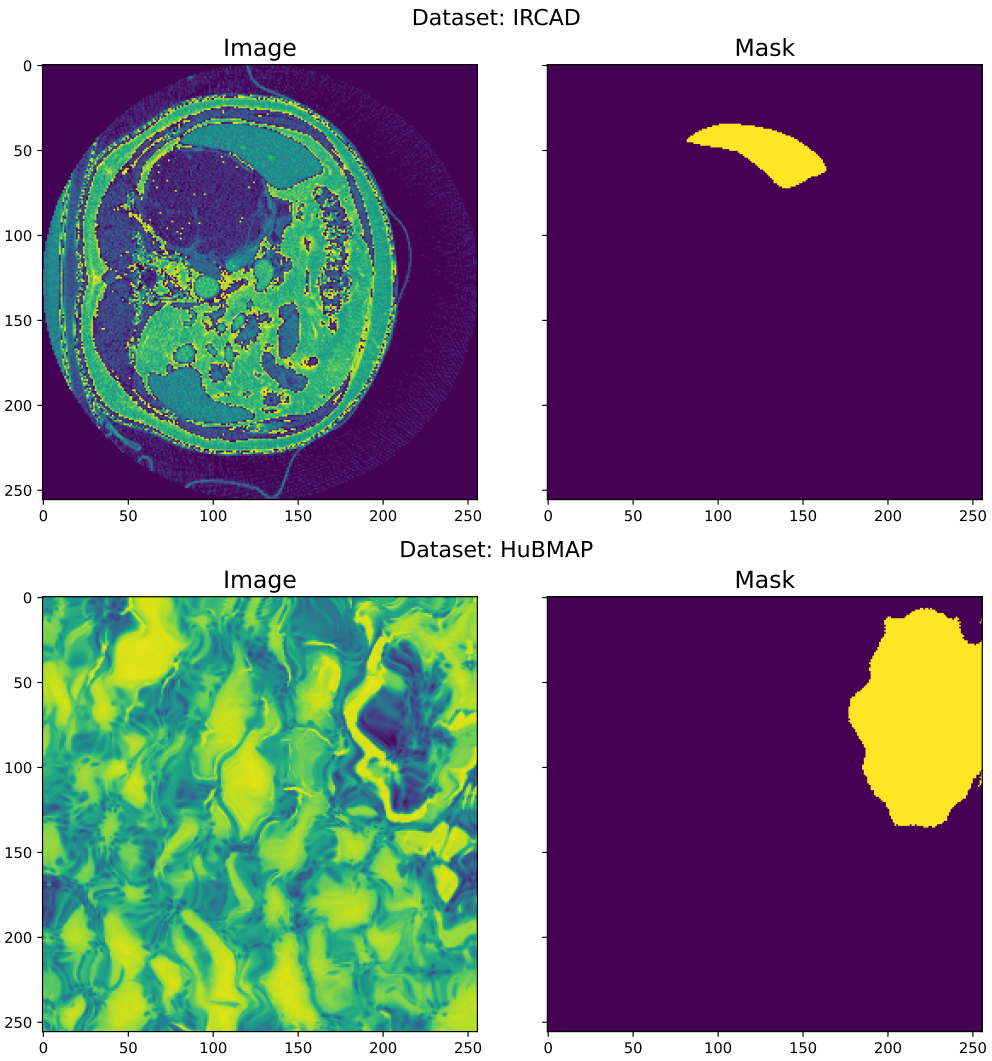

Figure 4.2: Examples of 256 × 256 tiles and the corresponding masks for the IRCAD and HuBMAP image segmentation datasets.

4.1.4 Experiment Setup & Evaluation

The experimental design was, in keeping with Chapter 3, kept relatively simple, with a focus on the question of whether the three confusion matrix based losses, the GMN, F1 and MCC, would be more effective than the three competitor loss functions used in this subsection: the binary cross entropy, on the basis that this is the standard loss used for classification, and the Dice and Focal losses, the two domain-specific losses introduced earlier.

The datasets were split into five folds, using a 20% validation and 20% test size (so that 60% of the data was used for training in each fold). However, care was taken to ensure tiles belonging to each original image were kept together. Given that both datasets contained 20 original (before tiling) images, a single fold, for either dataset, consists of 12 images for training, four for validation and four for testing. The U-Net model, a type of fully convolutional neural network, was chosen to be optimised since it was designed specifically for image segmentation [103].

For each fold and candidate loss function, a U-Net model with an encoder depth of four and a sigmoid activation function was trained with stochastic gradient descent with a batch size of 10, learning rate of 0.001 and momentum rate of 0.9 for a maximum of 300 epochs. After each training epoch, the model was assessed using the validation dataset. The weights from the epoch that achieved the highest validation MCC were then used for testing. The experiment is repeated 10 times with different initial model weights for each fold producing a total of 50 models trained with each loss.

Final testing performance was compared for the different training losses using accuracy, F1-Score, MCC, and the Intersection over Union (IoU) measure. The IoU, also know as the Jaccard index (J),

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4.11}$$

is commonly used to compare the similarity between two arbitrary shapes. The IoU considers the shape properties of the objects under comparison (e.g. the widths, heights, and localisations of two bounding boxes) and then calculates a normalised measure of overlap that focuses on their areas (or volumes) [104]. This makes the IoU invariant to the scale of the problem under consideration, an appealing property that has led to its popularity in the evaluation of image segmentation algorithms [105, 106].

4.1.5 Results & Discussion

Results: IRCAD Dataset. The average and standard deviations of the IoU, F1-Score, MCC and accuracy test scores are displayed in Table 4.1. The approximated GMN, F1 and MCC show a statistically significant improvement over the BCE, Dice and Focal competitor losses in terms of IoU, F1 and MCC at the 5% level. However, it is not possible in this experiment to distinguish the best-performing among the novel loss functions, as the results for GMN, F1, and MCC in Table 4.1 are not statistically significantly distinguishable, in spite of the F1-Score being less sensitive to the misclassification of negative examples. The explanation for better-than-expected performance of the F1 loss is likely to be that the asymmetry of the F1-Score with respect to positive and negative examples, which was noted in [52] as being disadvantageous for a performance measure, might in fact be beneficial for the loss derived from it. This is because the F1 loss would tend to focus more strongly on the minority class, rather than, as is the case for both the GMN and MCC losses, giving the two classes the same attention. Interestingly, it can be seen that the IoU (Figure 4.3) more clearly distinguishes between results obtained by the various

training losses than does the MCC (Figure 4.4); while the MCC can be used to derive a very effective loss function, and is a generally highly regarded metric, it may not be the best performance measure for the specific problem of image segmentation, with the domain-specific IoU being preferable. It can be easily seen from Table 4.1 that the binary cross entropy is the worst performing loss, perhaps highlighting that a vanilla loss function, without specific adaptations (as in Chapter 5) to handle class imbalance, cannot in general be expected to perform well in situations with challenging IRs. As a final note on this experiment, the results for the different loss functions are not statistically significantly distinguishable in terms of accuracy, but it is well known that this is a poor metric for evaluating performance under class imbalance, so an uninformative result might be expected.

Table 4.1: IRCAD test results.

Loss	IoU	F1-Score	MCC	Accuracy
BCE	0.492 (0.23)	0.598 (0.23)	0.637 (0.22)	0.969 (0.02)
Dice	0.577 (0.35)	0.633 (0.38)	0.636 (0.38)	0.972 (0.02)
Focal	0.589 (0.22)	0.676 (0.20)	0.685 (0.18)	0.953 (0.05)
GMN	0.692 (0.13)	0.742 (0.12)	0.747 (0.13)	0.975 (0.01)
F1	0.696 (0.14)	0.762 (0.14)	0.766 (0.14)	0.978 (0.01)
MCC	0.697 (0.17)	0.763 (0.16)	0.769 (0.16)	0.978 (0.02)

Results: HuBMAP Dataset. In relation to this dataset, in can be seen in Table 4.2, and in Figures 4.5 and 4.6, that results for the three novel loss functions unfortunately do not appear to be, and on investigation can be shown to be, statistically significantly better for the non-trivial performance measures (accuracy again being uninformative) than for the Dice and Focal losses. The performances of the five more classification-focused loss functions are equivalent up to 5% significance (using a paired t-test), with the only strong conclusion about them that can be drawn being that all five are statistically significantly better performing (as evidenced by

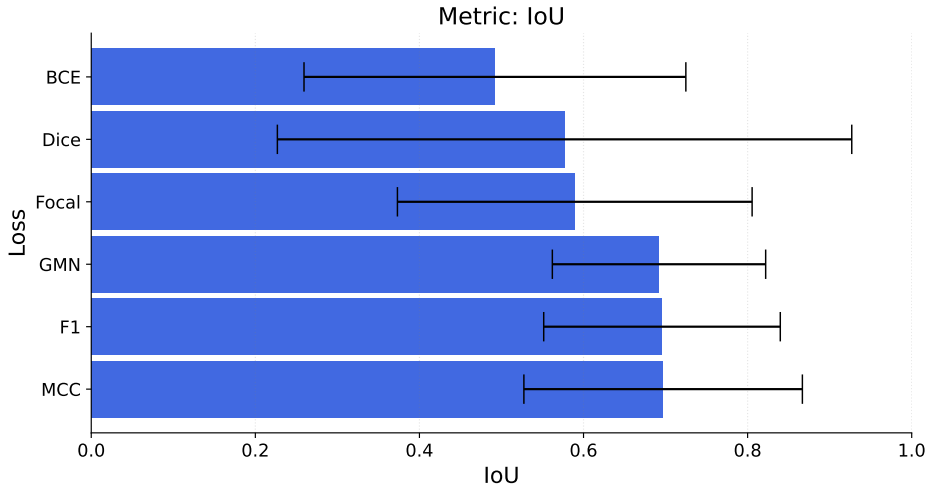

Figure 4.3: IRCAD Intersection over Union (IoU) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars.

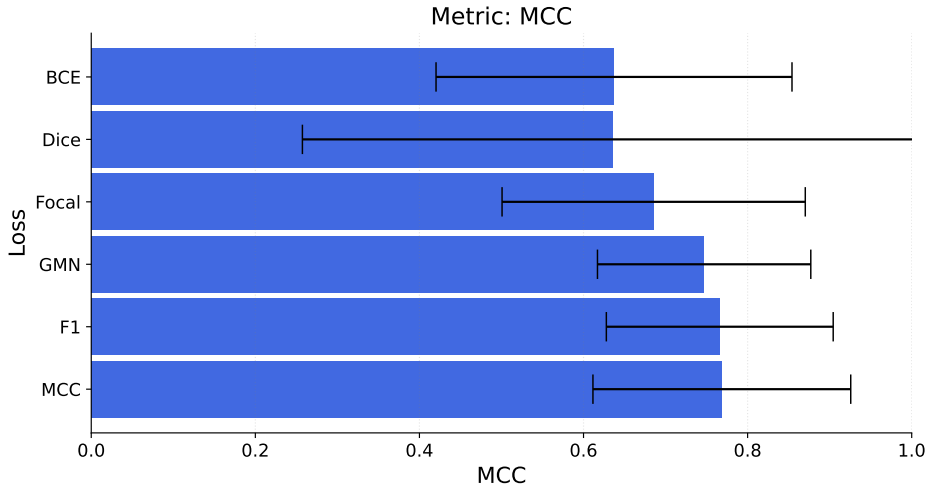

Figure 4.4: IRCAD Matthews Correlation Coefficient (MCC) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars.

a one-sided t-test) than binary cross entropy. The lack of superiority of the three novel loss functions, expected to outperform more substantially at higher IRs, was a surprise, as the HuBMAP dataset has an IR more than twice that of the IRCAD dataset, where the preferability of these three loss functions over the Dice and Focal losses had been demonstrated. However, there is more involved in assessing the difficulty of a classification problem than its IR, and it is possible that the HuBMAP

Table 4.2: HuBMAP test results.

Loss	IoU	F1-Score	MCC	Accuracy
BCE	0.550 (0.13)	0.618 (0.15)	0.653 (0.15)	0.995 (0.00)
Dice	0.663 (0.09)	0.725 (0.10)	0.726 (0.10)	0.994 (0.00)
Focal	0.665 (0.10)	0.728 (0.10)	0.729 (0.10)	0.993 (0.00)
GMN	0.68 (0.03)	0.721 (0.05)	0.715 (0.09)	0.992 (0.01)
F1	0.691 (0.02)	0.756 (0.02)	0.719 (0.09)	0.993 (0.00)
MCC	0.677 (0.09)	0.734 (0.09)	0.737 (0.09)	0.995 (0.00)

dataset had some unusual aspects. Notably, the standard deviation in HuBMAP scores compared to the IRCAD scores is in general much lower, for training using any of the loss functions considered. This implies initial weight settings have less effect on test performance, which was a surprise given the complexity of the U-Net model. Because of the possibility that there were issues with HuBMAP that went beyond the question of class imbalance, it would have been desirable to experiment with further image segmentation datasets, a topic which will be returned to in the discussion below.

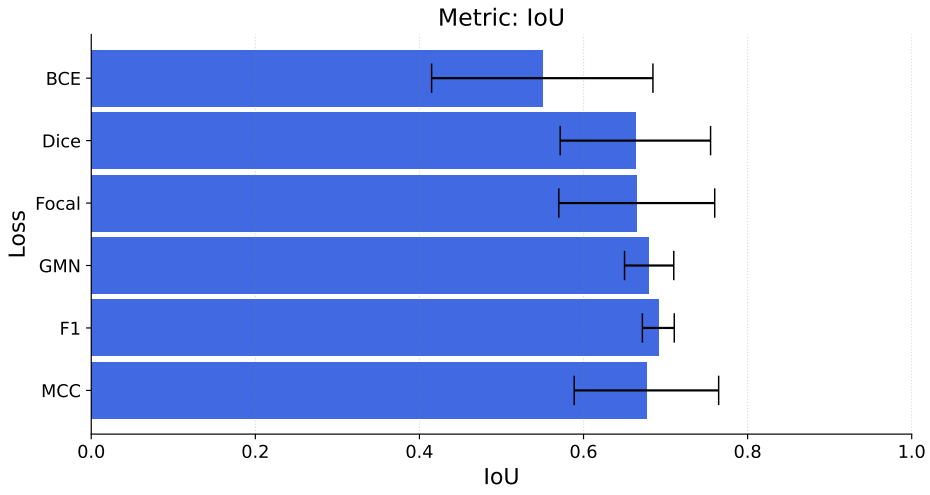

Figure 4.5: HuBMAP Intersection over Union (IoU) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars.

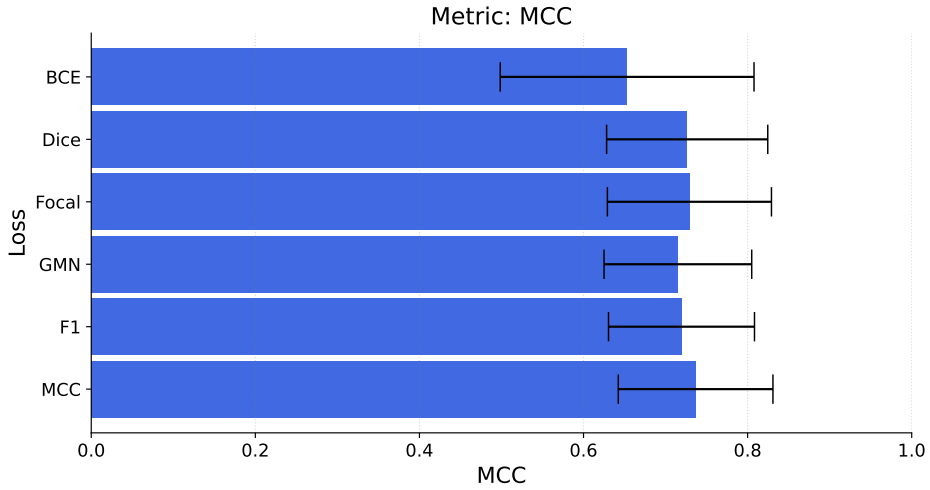

Figure 4.6: HuBMAP Matthews Correlation Coefficient (MCC) test results. Scores are averaged across the 5 k-fold splits; standard deviations are shown in black bars.

Conclusions. In summary, this section has demonstrated on two recent image segmentation datasets that loss functions such as the GMN, F1 and MCC, derived from approximated confusion matrix based performance measures, are at least competitive with, and for the case of the IRCAD dataset superior to, domain-specific losses (as evidenced in Tables 4.1 and 4.2). It has in addition been discovered that while the F1-Score has undesirable features for use as a performance measure (asymmetry between positive and negative examples, with correct classification of the former being deemed the more important), as was noted in Chapter 2, the loss function derived from it may work well precisely because of this asymmetry. However, it should be noted that it would not in Chapter 3 have been possible, unless one made a lucky choice of datasets, to have demonstrated the superiority of the GMN over the MSE and BCE using only two datasets, as were used in this chapter. Indeed, there were in Chapter 3 - at least until IRs higher than 50 were reached - isolated instances where one or other of the simpler loss functions produced a better result than the GMN. Clearly, therefore, the conclusions of this current section could be strengthened by experimentation on further image segmentation datasets,

especially ones with much higher IRs. This, however, would demand both a large number of additional very large datasets, and was not computationally feasible at the time of this work. Further investigations into extensions of the confusion matrix based loss function therefore took the route of considering multi-class extensions, this being also a much broader area of application than the very specialised area of image segmentation.

4.2 Multi-Class Classification

All of the previous work on classification-focused training metrics has so far focused on the *binary* classification problem (i.e. when there are just two classes, the majority and minority, typically with the minority class the one of interest). However, there are many examples of multi-class datasets which also present a form of the class-imbalance problem.

4.2.1 Class Imbalance in the Multi-Class Case

There would be many potentially complex ways to measure class imbalance in the multi-class setting, that attempt to characterise multiple aspects of the class distribution. However, the most common approach is to compare the ratio of the sizes of the smallest and largest sets, equivalent to the definition of the Imbalance Ratio (IR) for the binary case in equation 2.1; this definition will be repeated below for clarity,

$$IR = \frac{\max_i |C_i|}{\min_i |C_i|},\ i = 1..K, \tag{4.12}$$

where for C_i is the number of examples of class i.

4.2.2 Extension of Performance Measures to Multi-Class Classification

As discussed in Chapter 2, the confusion matrix is an excellent tool for assessing a classifier's performance as it can provide insight not only into the errors being made by a classifier, but, more importantly, the type of the resulting errors.

The 2×2 confusion matrix illustrated in Figure 2.1, however, considers only binary classification, where one of the labels is considered to be 'positive' and the other 'negative'. In the case of multi-class classification (i.e. the number of classes is greater than two), the dimension of the confusion matrix needs to be extended to $K \times K$, where K is the number of different classes, $C_1, C_2, ..., C_K$. Each column of the matrix represents the instances of a predicted class, while each row represents the instances of an actual class. An element $M(i, j)$ of the $K \times K$ confusion matrix M at row i and column j provides the number of instances for which the predicted class is j and the actual class is i. Such a multi-class confusion matrix is illustrated in Figure 4.7.

Figure 4.7: Multi-class classification problem confusion matrix

		Predicted Class			
		$\mathbf{C}_1$	$\mathbf{C}_2$	...	$\mathbf{C}_K$
Actual Class	$\mathbf{C}_1$	$M_{1,1}$	$M_{1,2}$	...	$M_{1,K}$
	$\mathbf{C}_2$	$M_{2,1}$	$M_{2,2}$	...	$M_{2,K}$
	...	...	...	...	...
	$\mathbf{C}_K$	$M_{K,1}$	$M_{K,2}$	...	$M_{K,K}$

Clearly, the characterisation of TP, TN, FP, and FN instances needs to be extended. This can be done by decomposing the multi-class confusion matrix into set of binary confusion matrices (or contingency tables) with respect to each class C_i. Let C_+ be C_i and let C_- be all classes except for C_i, i.e. $C_- = \{C_j\} - C_i$ (for all classes except i). A binary confusion matrix can then be constructed from C_+ and C_-, from which TN_i, FN_i, FP_i and TP_i can be calculated as below:

$$
\begin{aligned}
TP_k &= M(k,k),\\
TN_k &= \sum_{i\neq k}\sum_{j\neq k} M(i,j),\\
FP_k &= \sum_{i\neq k} M(i,k),\\
FN_k &= \sum_{j\neq k} M(k,j).
\end{aligned}
\tag{4.13}
$$

Performance metrics for each class C_i can then be constructed in the exact fashion described in Chapter 2. However, since these measures are calculated for each class separately, they need to be combined in some way to order to create a single-number performance measure. There are two approaches to this, referred to as *micro* and *macro* averaging.

When using micro-averaging, each sample is weighted equally and the TP, FN, FP, TP contributions for each class are combined globally, as

$$
\begin{aligned}
TP &= \sum_{k=1}^{K} TP_k,\\
FP &= \sum_{k=1}^{K} FP_k,\\
FN &= \sum_{k=1}^{K} FN_k,\\
TN &= \sum_{k=1}^{K} TN_k,
\end{aligned}
\tag{4.14}
$$

with the desired measure then computed from these summed terms. Micro-averaging then aggregates the contributions of all classes to compute an average metric. The micro-averaged Precision and Recall can be expressed as

$$
Recall_{micro} = \frac{TP}{TP+FP} = \frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FP_k}, \tag{4.15}
$$

$$Precision_{micro} = \frac{TP}{TP+FN} = \frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FN_k}. \tag{4.16}$$

Micro-averaged approximations for the G-Mean, F1 and MCC can be equivalently constructed as:

$$G\text{-}Mean_{micro} = \sqrt{\frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FN_k} \times \frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FP_k}}, \tag{4.17}$$

$$F1\text{-}Score_{micro} = \frac{2\sum_k TP_k}{2\sum_k TP_k + \sum_k FP_k + \sum_k FN_k}, \tag{4.18}$$

$$MCC_{micro} = \frac{\sum_k TP_k \times \sum_k TN_k - \sum_k FP_k \times \sum_k FN_k}{\sqrt{(\sum_k TP_k + \sum_k FP_k)(\sum_k TP_k + \sum_k FN_k)(\sum_k TN_k + \sum_k FP_k)(\sum_k TN_k + \sum_k FN_k)}}. \tag{4.19}$$

Macro-averaging, in contrast, is similar to the 'one-vs-all' classification technique. The chosen metric is first computed independently, for each C_i, and then averaged, weighting each class equally. To compute the macro-averaged Precision and Recall, local metrics are calculated first for each class and then averaged:

$$Recall_k = \frac{TP_k}{TP_k + FP_k}, \tag{4.20}$$

$$Precision_k = \frac{TP_k}{TP_k + FN_k}, \tag{4.21}$$

$$Recall_{macro} = \frac{1}{K}\sum_i Recall_i, \tag{4.22}$$

$$Precision_{macro} = \frac{1}{K}\sum_{i} Precision_i. \tag{4.23}$$

Similarly, macro-averaged approximations for the G-Mean, F1 and MCC can be constructed:

$$G\text{-}Mean_{macro} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{TP_k}{TP_k+FN_k}\times\frac{TP_k}{TP_k+FP_k}}, \tag{4.24}$$

$$F1\text{-}Score_{macro} = \frac{1}{K}\sum_{k=1}^{K}\frac{2TP_k}{2TP_k+FP_k+FN_k}, \tag{4.25}$$

$$MCC_{macro} = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k\times TN_k - FP_k\times FN_k}{\sqrt{(TP_k+FP_k)(TP_k+FN_k)(TN_k+FP_k)(TN_k+FN_k)}}. \tag{4.26}$$

If there are an equal number of samples for each class $i = 1..K$, then macro and micro averaging will result in the same score. However, in the current context, this will plainly not be the case, and hence it would be expected that micro and macro averaging would here behave quite differently and could give different results.

A further definition of the Matthews Correlation Coefficient (MCC) for use in a multi-class scenario was proposed by [107]. It is the most frequently used form of the multi-class MCC in machine learning, and is built into many packages. However, it is intended for performance measurement, not as the basis for a loss function, for which it would be disadvantageous due to the complexity of the derivatives. It is used here only to compare the performance of the various models in the experiments to follow, on the basis that it is the most common scoring metric in the literature. This form of the MCC is constructed as the discretised form of the R_K coefficient, which itself is a generalisation of the Pearson's correlation coefficient, This form of

the MCC is constructed as the discretised form of the R_K coefficient [108], which captures the relationship between two variables, similar to Pearson's correlation, but extends its applicability to ordinal and categorical data,

$$MCC = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}}, \tag{4.27}$$

where we define

$$\begin{aligned} t_k &= \sum_i^K C_{ik}, \\ p_k &= \sum_i^K C_{ki}, \\ c &= \sum_k^K C_{kk}, \\ s &= \sum_i^K \sum_j^K C_{ij}. \end{aligned} \tag{4.28}$$

4.2.3 Construction of Approximated Multi-Class Training Metrics

In order to construct an approximated version of the multi-class confusion matrix M, it is required that labels are one-hot encoded and model outputs are kept as class probabilities (i.e. by applying softmax). Then we have

$$M_{apx} = \hat{\mathbf{y}}^\mathsf{T} \cdot \mathbf{y}, \tag{4.29}$$

where $\hat{\mathbf{y}}$ is the matrix of class probabilities and $\mathbf{y}$ is the one-hot encoded targets, respectively. Subsequently, approximated equivalents to all of the previously defined multi-class measures can be defined, and suitable novel loss functions derived from approximated multi-class performance metrics.

Considering the choice between micro- and macro-averaged forms of the ap-

proximated loss functions, after initial experimentation the macro-averaged variants were preferred. For macro-averaging, averaging approaches other than the arithmetic mean, such as the geometric or harmonic mean, were also explored, although were found both to be overly pessimistic and so were not pursued further.

4.2.4 Long-Tailed Recognition

A multi-class dataset where a few classes have significantly more data than many other classes is said to follow a long-tailed class distribution. Numerous applications require or even prioritise good performance on these rare classes. For example, for reasons of driving safety, autonomous vehicles should recognize not only common objects such as cars and pedestrians, but also rare ones like baby strollers and animals [109]. Other examples include species classification [110], facial recognition [111], medical image diagnosis [19] and unmanned aerial vehicle detection [4]. However, classifier models trained using standard approaches on these types of long-tailed datasets can become easily biased towards the majority classes, in the same way that binary classifiers can make over-assignments to the larger of the two classes, leading to poor model performance and a real difficulty in obtaining performance on the tail classes that have limited data.

The challenge of training a model on a long-tailed dataset with many classes, which achieves high accuracy when averaged across all classes, is known as the *long-tailed recognition* (LTR) problem [7]. LTR can be considered as a more specific and challenging sub-task within class-imbalanced learning, where there are a large number of classes and the tail-class samples are often very scarce [4]. Being relevant to many practical areas, including various computer vision tasks such as image classification, detection and segmentation [4], the LTR problem has attracted increasing attention, with researchers in recent years proposing many approaches to alleviate the issue and improve performance, particularly using deep neural networks.

Long-tailed recognition is challenging due to two main difficulties. Firstly, data imbalance across classes makes deep models biased toward head (frequently seen) classes so they consequently perform poorly on tail classes; and, secondly, a lack of tail-class samples makes it even more challenging to train models for tail-class classification [4]. It should also be noted that while LTR is related to few-shot-learning [112, 113], which aims to train models with a limited number of labelled samples (e.g. 1 or 5), few-shot learning can be regarded as a sub-task of long-tailed learning, in which the tail classes generally have a very small number of samples [4].

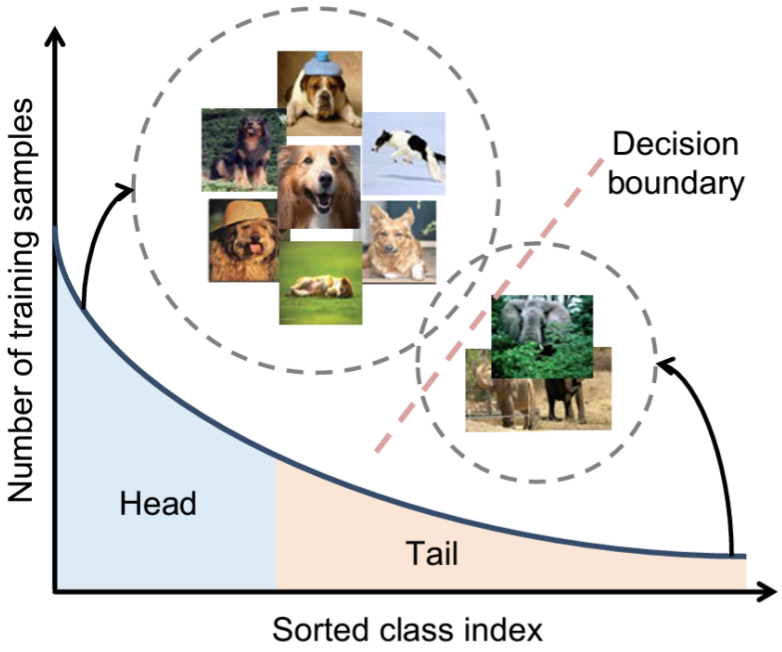

Figure 4.8: Label distribution of a long-tailed image dataset. Original image from [4]

.

The proposed methods for handling LTR can be grouped into five main categories, each of which will be explained below: data manipulation, loss function engineering (which will be the core approach of this chapter), two-stage methods, ensembling, and parameter regularisation. It should be noted that the literature relating to long-tailed recognition is large and that there is some overlap of these

approaches with those discussed for binary problems in Chapter 2. As such, the following brief review of standard methodologies for LTR is limited only to those used for comparison purposes in this section of this chapter.

Data manipulation (re-balancing) approaches encompass methods such as the under- and over sampling described for the case of binary class imbalance in Chapter 2. Another form of re-balancing technique is to generate augmented samples to supplement tail classes. Examples include SMOTE (again, introduced in Chapter 2) and major-to-minor translation (M2m) [114]. However, these approaches to LTR suffer from the same issues as in the binary case, when imbalance is high, namely that over-sampling degrades performance on the majority classes, under-sampling methods degrade the generalisation of models and data augmentation methods are expensive to develop [4].

Loss manipulation (re-weighting) involves a change of the loss function used for training from one that is generic (for example, cross entropy (CE), to one that is better suited to LTR, either by adapting existing losses with a weighting scheme or by proposing entirely novel loss functions. The assignment of weights to classes or training examples aims to modify their gradients to make class-imbalanced data contribute more equally to training [4]. Examples of adaptations to existing loss functions include the class balanced loss (CB) [6], logit-adjusted loss [115], focal loss [99], and label-distribution-aware margin (LDAM) loss [7]. Novel losses already proposed in the literature include the equalisation loss (EQ) [116], distributional robustness loss (DRO) [117] and the parametric contrastive loss (PaCo) [6].

Two-stage methods have found benefits for LTR in de-coupling the training process into feature and classifier learning [118]. In the first stage, the feature learning stage, the model is trained with a standard loss, and in the second stage the model is re-trained (while freezing deeper layers in the architecture) with a

class-balancing loss. These methods assume that the bias towards the majority classes is significant only in the classifier (i.e. the last fully connected layer of the network), or that tweaking the classifier layer can correct the underlying biases in the feature representation. Examples of two-stage approaches include self-supervised to distillation (SSD) [119] and bilateral-branch networks (BBN) [120].

Ensembling methods focus on generating a balanced model by assembling and grouping models that may individually be focused on subsets of classes. Typically, classes are separated into groups, either automatically within the algorithm, or as a pre-training decision, where classes that contain similar training instances are grouped together. Individual models are then allowed to focus on each such group and combined to form a multi-expert framework [117]. Ensemble methods have been shown to improve generalisation on the long tailed recognition problem. Examples include approaches such as RIDE [121] and ACE [122].

Parameter regularisation is primarily a technique to help with overfitting and improve generalisation, usually by adding extra terms to the loss function to solve an ill-posed training problem, though it can also be done post-hoc. Reduction of the sizes of weights can help to avoid an unwanted developing (or developed, in the post-hoc case) focus on the majority class, manifested by large weights. It is arguably less explored in the literature for LTR as a method for handling class imbalance than those methods previously listed; however, recent work by [5] demonstrated that the use of weight decay [123] and MaxNorm constraints [124] during training, as well L2- and τ normalisation [118] (here used as a post-hoc approach that after training reduces the sizes of the weights in the outer layers), can significantly improve performance.

From recent work in LTR, the most relevant observations for the LTR experiments conducted in this chapter are that LTR can be better addressed by de-coupling feature learning and classifier learning, rather than training them jointly [118], and

the use of parameter regularisation during training to prevent model weights from growing too fast and developing a bias to (training) majority classes [5].

4.2.5 CIFAR100-LT Dataset

The benefit of using the novel multi-class approximated confusion matrix based training metrics introduced earlier in this section will now be explored on long-tailed versions of the popular CIFAR100 dataset [125].

In line with the literature and following [126], we modify the CIFAR100 dataset, a balanced 100-class image classification dataset, by downsampling per-class training examples using an exponential decay function to create a long-tailed version. The resulting downsampled datasets, referred to as CIFAR100-LT, maintain 100 classes and a balanced validation set for evaluation. However, they feature distinct long-tailed training sets depending on the chosen imbalance ratio (IR) for downsampling. In this experiment, two IR levels are selected: 100 and 500, yielding two different long-tailed training sets, as illustrated in Figure 4.9.

It is important to note that an explicit out-of-sample test set is not used, as CIFAR100 only contains a single train/val split. Using the validation set during training is a standard approach in the literature [127]. The balanced validation/test dataset is adopted to ensure that 1) top-1 accuracy serves as a reliable evaluation metric, and 2) minority training examples are adequately represented. Furthermore, no hyperparameter optimization is conducted on the validation set in the current experiment, allowing it to be legitimately considered and referred to as the test set from this point onward.

4.2.6 Experiment Setup & Evaluation

On the basis of the earlier discussion of the long-tailed learning literature, which recommended a two-stage training process, such a process will be adopted in this

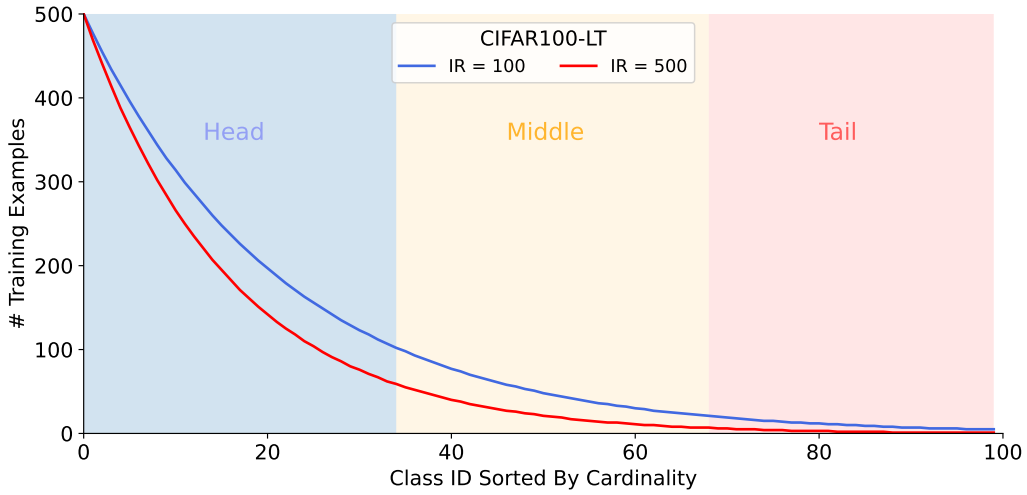

Figure 4.9: CIFAR100-LT train set frequency distributions for allocated class cardinalities for the different target imbalance ratios (IRs), showing also the division of classes into Head, Middle, and Tail class frequencies.

work, with the goal of decoupling feature learning and classifier learning, as first suggested by Kang et al. [118]. Feature learning, the first stage of the two-stage process, is here performed using a standard cross entropy (CE) loss function, with the addition of weight decay, as used in [5]. Weight decay is a well-studied technique [128, 129] used to constrain a network by limiting the growth of the network weights, in this case adding an L2 penalty to the loss function, which decreases the complexity of the network, effectively mitigating over-fitting and improving generalisation by encouraging weights to grow within a norm ball and, conversely, discouraging them when their norms exceed the radius. The behaviour of the model can then modified in one of two ways, the first being subsequent training with a classification-focused loss, the second - for the purpose of comparison, as it is a well-regarded method in the LTR community - being post-hoc weight regularisation; these will be discussed further below.

Performance assessment in this experiment is performed in line with the literature on long-tailed recognition. In line, again, with usual practices in LTR, we

train on the long-tailed class-imbalanced training set and evaluate on its (balanced) test set. To allow comparison to other benchmark studies, test set evaluation is performed using the Top-1 classification accuracy, which is essentially the equivalent of standard accuracy for a binary problem and, as such, is potentially subject to the same criticism of being misleading in instances of high IR. However, it should be noted that, although the training set is class-imbalanced (deliberately undersampled), the test set is balanced, so in this case the accuracy *is* in fact a reasonable measure of overall performance. As well as the Top-1 Accuracy, per-class accuracy is measured on three disjoint subsets - Head classes (classes each with over 100 training samples), Middle classes (classes with between 20 to 100 training samples), and Tail classes (classes with under 20 training examples). This helps to understand the detailed characteristics of each method in the presence of class imbalance. As well as using Top-1 Accuracy to assess test performance, this study will additionally use the Matthews Correlation Coefficient, using the third of the multi-class versions of the MCC, that form given by Equation 4.27, as this provides a simple, single-number performance assessment for multi-class problems, as such being implemented, for example, in the scikit-learn and PyTorch packages.

In line with other studies [5, 4] (noting that these are available only for the lower IRs used in this study) the network architecture chosen for these experiments is ResNet32 [130]. Stochastic Gradient Descent (SGD) is used for optimisation, with batch size 64, momentum 0.9, and a cosine learning rate scheduler [131] which gradually decays learning rates from 0.01 to 0. The baseline first-stage model is trained in a similar fashion to [5], the ResNet32 model being trained from scratch for 200 epochs, using weight decay (WD) regularisation (as noted above). The baseline model (naive model, trained using a cross entropy (CE) loss, plus weight decay) is then, again as noted above, augmented by one method, selected from one of two

categories of intervention. The first category is post-hoc and comprises of L2 or τ normalisation. The second category, used alternatively, rather than additionally, to methods from the first, modifies the training process for the unfrozen outer layer, and comprises of a selection from one three confusion matrix based loss functions (GMN, MCC, or F1), or the use of one competitor loss, namely the class-balanced cross entropy loss (CB) [6], label-distribution-aware margin (LDAM) loss [7] and the focal loss with (Focal+CB) and without (Focal) class-balancing.

The primary goal is to evaluate the benefit of using a novel multi class confusion matrix based loss function for classifier (i.e. second stage) learning over a number of other proposed second-stage loss functions. However, it is important also to benchmark the results against the recent work of others, as summarised in Table 4.3. Among the results cited in this table, those of one work will be of particular significance for comparison, this work being the recent paper of [5], whose results using a hybrid methodology (bottom line in Table 4.3) represent the current state of the art.

4.2.7 Results & Discussion

The results discussion is divided into two parts based on the IRs of the two CIFAR100-LT datasets considered, their imbalance ratios of 100 and 500, respectively. For the IR=100 dataset, test performances can be compared to other LTR studies (their results are shown in Table 4.3); however this cannot be done for the higher imbalance ratio, since results for an IR of 500 are not, to the author's knowledge, available in any other works. Results using competitor loss functions given in Tables 4.4 and 4.5 are generated here for the purpose of comparison, rather than being reproduced from elsewhere (which, as noted, would in any case not be possible for the higher IR).

CIFAR100-LT (IR = 100). Table 4.4 gives the test Top-1 Accuracies for the CIFAR100 dataset with an IR of 100. In order to give a deeper insight into the results,

Table 4.3: Benchmark CIFAR100-LT (IR = 100) test Top-1 Accuracies.

CIFAR100-LT IR: 100		
Approach	Top-1 Acc. (%)	Description
CE	38.32	Cross entropy loss [6]
Data re-balancing		
ReSample	33.44	Oversampling [6]
SMOTE	34.01	Synthetic minority oversampling technique [114]
M2m	42.90	Major to minor translation (M2m) [114]
Loss manipulation		
CE+CB	39.60	Class balanced cross entropy loss [6]
Focal	38.41	Focal loss [75]
Focal+CB	39.60	Class-balanced focal loss [6]
LogitAdjust	42.01	Logit-adjusted cross entropy loss [115]
LDAM	42.04	Label-distribution-aware margin (LDAM) loss [7]
EQLoss	43.38	Equalisation loss [116]
DRO-LT	47.31	Distributional-robustness loss [117]
PaCo	52.00	Parametric contrastive learning [132]
Two-stage/de-coupling methods		
BBN	42.56	Bilateral branch network [120]
DiVE	45.35	Distill the virtual examples [133]
SSD	46.00	Self-supervised to distillation [119]
Ensembling		
ACE (4-expert)	49.60	Ally complementary experts [122]
RIDE (4-expert)	49.10	Routing diverse distribution-aware experts [121]
Regularisation		
τ-norm	47.73	(Post-hoc) τ normalisation [118]
WD+L2-norm	49.60	WD + (post-hoc) L2 normalisation [5]
WD+τ-norm	51.31	WD + (post-hoc) τ normalisation [5]
Loss manipulation, de-coupling, regularisation		
Hybrid	53.35	CB + WD + MaxNorm ([5]

the test accuracies are separated into the Head, Middle and Tail class frequencies described in the previous section, and 'All' which represents the Top-1 Accuracy over

the whole dataset (as the standard metric that is quoted in the LTR literature). It is clear from this table that two-stage training is preferable, as an addition to the baseline model, to the use of regularisation. It is also clearly evident that for the lower class frequencies (Middle and Tail) the three novel loss functions represent a significant improvement on the state of the art. For the Head class frequency, however, the most effective second-stage loss function is standard cross entropy (CE); this result is not unexpected given that focus on minority classes inevitably impacts classification of the majority, such that for more balanced datasets a conventional choice of loss may be preferred. It should also be noted that the outperformance of CE for the Head class does not impact on the conclusion that the overall (final column of Table 4.4) winning second-stage loss functions are the three novel ones of this chapter (these three not being statistically significantly distinguishable, when all class frequencies are combined). Figure 4.11 provides a visual summary of the results; while the (CE+CB) method does well, and appears to come close to the performance of the (GMN, F1, MCC) group, it is even so statistically significantly beaten by use of this group of novel loss functions, due mostly to the relative underperformance of (CE+CB) on the Head class. Finally, Figure 4.12 looks in detail at the Top-1 Accuracies for all the classes considered, rather than grouping them into (Head, Middle, Tail). comparing the baseline model (CE+WD, no second-stage training) with second-stage training using the MCC loss. It is evident from this figure that while performance drops off very noticeably for the baseline model as class ID increases (class frequency decreases), this degradation of performance is much less obvious for the model with second-stage MCC training.

Test Top-1 Accuracy: IR=100				
	Train Class Frequency			
Model	Head	Middle	Tail	All
First-stage (feature) training, method as in [5]				
CE	77.30	46.80	14.80	47.90
As above, plus post-hoc regularisation [5] (one choice from the below)				
L2-norm	76.09	47.74	20.87	49.60
τ-norm	73.11	47.69	30.10	51.31
As first line, plus second-stage (classifier) training (one choice from the below)				
CE	**78.13 (0.17)**	41.49 (0.50)	4.04 (0.94)	43.08 (0.22)
Focal	77.52 (0.22)	44.08 (0.50)	9.56 (1.07)	45.43 (0.24)
LDAM	77.07 (0.14)	47.47 (1.47)	18.47 (0.47)	49.13 (0.61)
Focal+CB	75.84 (0.30)	50.76 (0.39)	26.48 (0.92)	52.25 (0.25)
CE+CB	69.81 (0.65)	50.23 (0.58)	37.57 (0.64)	53.29 (0.18)
GMN	71.45 (0.61)	**52.35 (0.84)**	38.35 (0.96)	**54.18 (0.33)**
F1	72.60 (0.55)	**52.14 (0.66)**	37.71 (1.13)	**54.27 (0.32)**
MCC	73.44 (2.68)	**52.14 (0.85)**	**41.34 (3.04)**	**54.28 (0.42)**

Table 4.4: Top-1 Accuracy on CIFAR100-LT (IR=100) test set, for baseline model plus weight decay (as in [5]), with various choices of subsequent intervention, broken down into regularisation (as in [5]) and second-stage classifier training, the latter including both a loss used in [5] (CE+CB) and three novel loss functions of this thesis (GMN, F1, MCC). "CB": class-balancing [6]; "LDAM" label-distribution-aware margin (LDAM) loss [7]. Results are split vertically by training methodology and horizontally by train class frequency, with standard deviations in brackets, and bolding indicating a win or tie at 5% statistical significance among the second-stage classifier models.

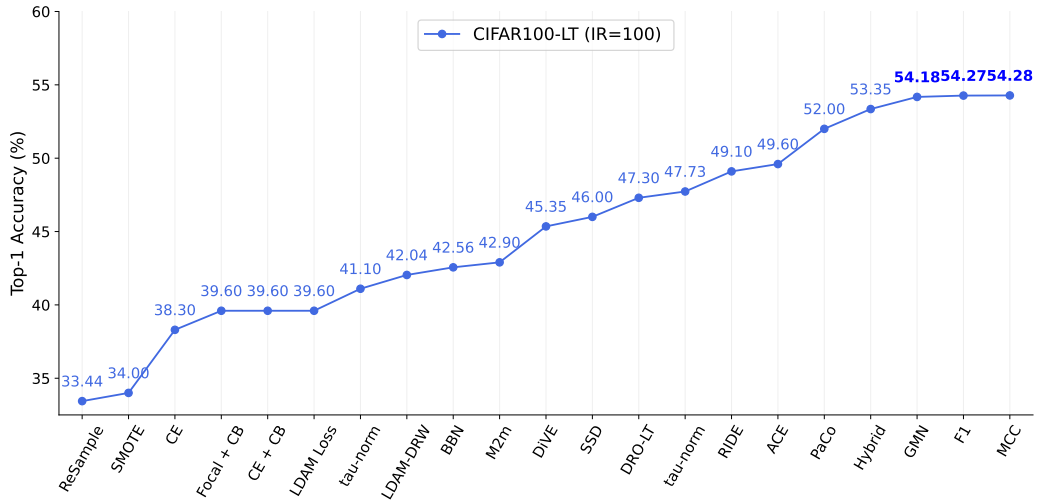

Figure 4.10: CIFAR100-LT (IR=100): Comparison of average Top-1 Accuracies of the confusion matrix based loss functions used in this chapter (GMN, F1 & MCC losses, displayed in bold) to recent state-of-the art results for this dataset.

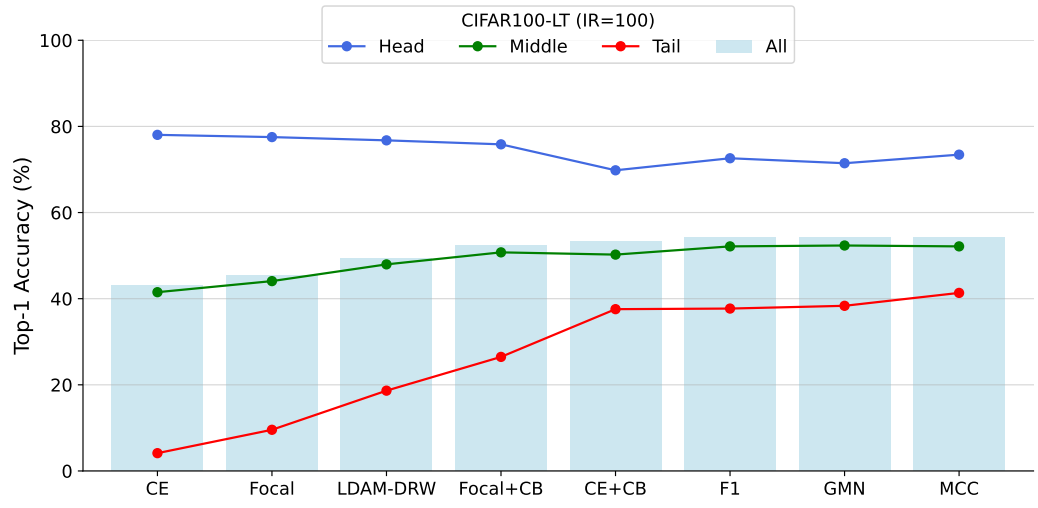

Figure 4.11: Comparison of Top-1 Accuracies on CIFAR100-LT (IR=100) across the second-stage losses considered, for differing train class frequencies.

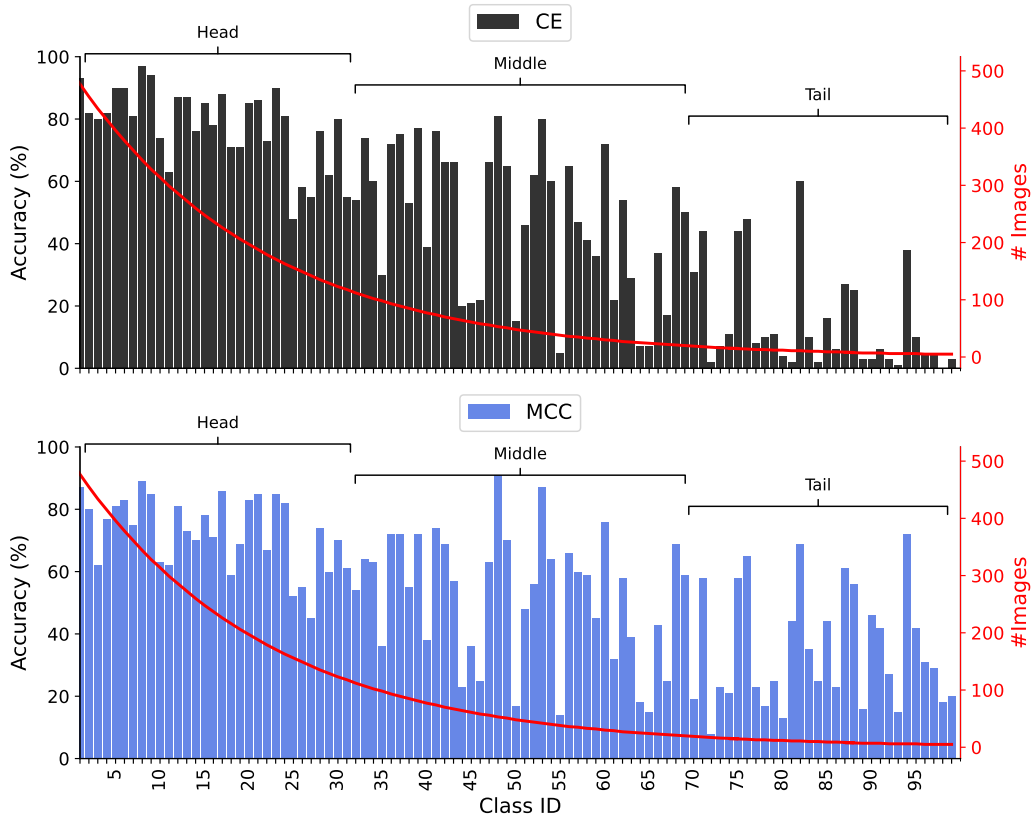

Figure 4.12: CIFAR100-LT (IR=100) per-class test Top-1 Accuracy for second-stage model trained with MCC loss compared to per-class test Top-1 Accuracy for baseline model with CE loss and no second-stage training. Red line indicates class count of training dataset (test dataset is balanced).

CIFAR100-LT (IR = 500). Table 4.5 gives equivalent results to Table 4.4 for the IR=500 dataset. Comparing the performances of the different second-stage loss functions, the results are broadly similar to those for IR=100, except that for the Middle and Tail classes F1 and GMN outperform (to 5% significance) the MCC loss, as well as all other considered losses. It is, however, the MCC that is the overall winner, as the underperformance for the Head class of the MCC loss, relative to that of the Head class winner, CE, is less than for the F1 and GMN losses. Equivalently to Figure 4.11, Figure 4.13 gives a visual presentation of the results, where it can be seen that the relative outperformance of (F1, GMN) on the Tail class, compared to the worst-performing second-stage loss function, Focal, is even more striking.

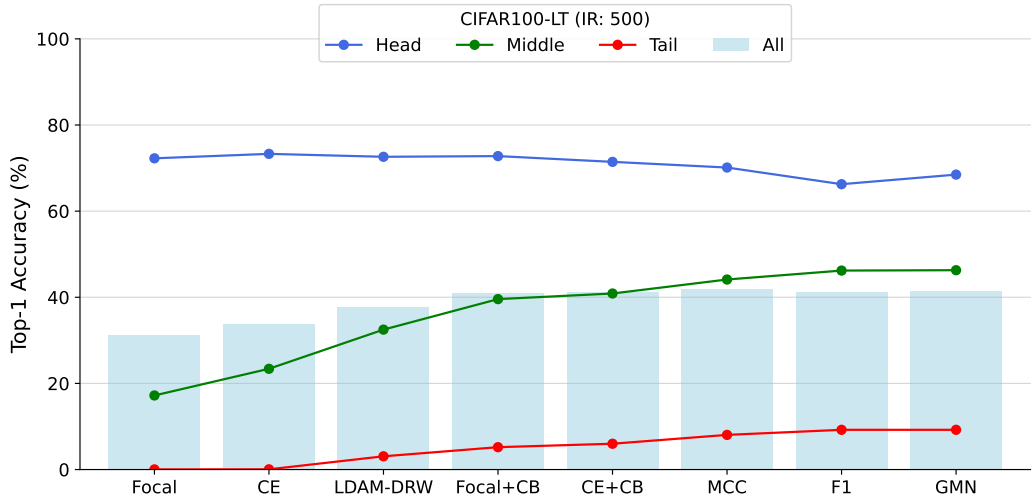

Figure 4.13: Comparison of Top-1 Accuracies on CIFAR100-LT (IR=500) across the second-stage losses considered, for differing train class frequencies.

Comparison of Results for IRs of 100 and 500. Table 4.6 compares the performance of the difference methods, now across the two IRs considered, using the MCC of 4.27 as a performance measure. Again, it is clear that two-stage training is a better-performing methodology than the use of post-hoc regularisation. Comparing the different second-stage loss functions, for IR = 100, GMN, F1, and MCC tie (to 5% significance), and for IR = 500 the winner is the MCC loss.

Test Top-1 Accuracy: IR=500				
	Train Class Frequency			
Model	Head	Middle	Tail	All
First-stage (feature) training, method as in [5]				
CE	73.90	30.3	1.40	36.90
As above, plus post-hoc regularisation [5] (one choice from the below)				
L2-norm	73.61	34.60	3.9	39.01
τ-norm	54.10	24.72	13.51	31.70
As first line, plus second-stage (classifier) training (one choice from the below)				
CE	**73.30 (0.14)**	23.39 (0.49)	0.05 (0.09)	33.86 (0.20)
Focal	72.26 (0.56)	17.20 (0.91)	0.03 (0.07)	31.32 (0.47)
LDAM-DRW	72.62 (0.95)	32.49 (0.80)	3.07 (0.72)	37.71 (0.76)
Focal+CB	72.78 (0.55)	39.57 (0.41)	5.19 (0.36)	40.88 (0.09)
CE+CB	71.44 (0.77)	40.87 (0.30)	5.99 (0.63)	41.11 (0.25)
F1	66.26 (1.07)	**46.20 (0.75)**	**9.22 (0.82)**	41.27 (0.23)
GMN	68.48 (1.75)	**46.29 (1.63)**	**9.22 (0.71)**	41.37 (0.26)
MCC	70.12 (1.10)	44.12 (0.40)	8.05 (0.77)	**41.86 (0.30)**

Table 4.5: Top-1 Accuracy on CIFAR100-LT (IR=500) test set, for baseline model plus weight decay (as in [5]), with various choices of subsequent intervention, broken down into regularisation (as in [5]) and second-stage classifier training, the latter including both a loss used in [5] (CE+CB) and three novel loss functions of this thesis (GMN, F1, MCC). "CB": class-balancing [6]; "LDAM" label-distribution-aware margin (LDAM) loss [7]. Results are split vertically by training methodology and horizontally by train class frequency, with standard deviations in brackets, and bolding indicating a win or tie at 5% statistical significance among the second-stage classifier models.

Table 4.6: Overall (all class frequencies) Matthews Correlation Coefficient (MCC) scores on CIFAR100-LT for the two imbalance ratios (IRs).

Imbalance Ratio (IR)	100	500
First-stage (feature) training, method as in [5]		
CE	46.78	36.25
As above, plus post-hoc regularisation [5] (one choice from the below)		
L2-norm	50.04	38.32
τ-norm	51.77	31.04
As first line, plus second-stage (classifier) training (one choice from the below)		
CE	42.76 (0.23)	33.56 (0.20)
Focal	45.11 (0.25)	31.02 (0.47)
LDAM-DRW	48.91 (0.40)	37.33 (0.78)
Focal+CB	51.87 (0.24)	40.42 (0.08)
CE+CB	52.87 (0.18)	40.63 (0.25)
GMN	**53.74 (0.34)**	40.88 (0.27)
F1	**53.85 (0.32)**	40.77 (0.23)
MCC	**53.85 (0.41)**	**41.38 (0.30)**

4.3 Discussion

This chapter examined alternatives (F1 and MCC losses) to the GMN loss introduced in Chapter 3.4, and the extension of the classification focused loss methodology to the case of multi-class problems.

Firstly, it was demonstrated that, with the underlying concept of the approximated confusion matrix, a differential approximation to the standard confusion matrix, any confusion matrix-oriented performance metric can, in principle, be converted into a loss function suitable for gradient based neural network training. Two such new losses were introduced: the F1 loss, based on the *approximated* F1-Score, and the MCC loss, based on the *approximated* Matthews Correlation Coefficient. The MCC, F1 and the (previously introduced) GMN loss were applied to two recent image segmentation datasets. Image segmentation, a sub-task of the broader field of computer vision, allows the practical demonstration of novel loss functions using deep network architectures, in this case applying a U-Net architecture, a commonly used model for image segmentation tasks. The relative benefits of the approximated measures were clear when they were compared to the use of a standard loss (binary cross entropy), but they also outperformed the use of two recently introduced image-classification-focused losses, the Dice loss [99] and the Focal loss [75], relative to both domain-specific measures such as the Intersection over Union (IoU) and traditional class-imbalance measures such as the MCC and F1-Score.

Secondly, having demonstrated convincing results on (real-world) binary class-imbalanced datasets, the concept of the approximated confusion matrix was extended to the multi-class setting, having first considered the different ways in which classification performance can be measured when working with multiple class labels, including the concept of micro and macro averaging. Both micro and macro forms of approximated performance measures, such as the G-Mean or F1, are possible,

but as initial investigations indicated that macro averaging was the more tractable approach, and so the macro-averaged forms of the measures were therefore adopted for the remainder of the multi-class work. Three macro-averaged approximated loss functions, the GMN_{macro}, $F1_{macro}$, and MCC_{macro} were then used as candidate loss functions for the second stage, or *classifier* learning stage, of a two-stage approach to a common multi-class class-imbalanced task known as the Long Tailed Recognition (LTR) problem. State-of-the art results were achieved overall (across all categories of class frequency), with the (GMN, F1, MCC) losses outperforming a recent work [5] that had represented a considerable advance, with the predominant advantage being seen, as might be expected, for the least frequent (Tail) classes.

It has thus been shown, in this and the previous chapter, that classification focused loss functions are potentially very powerful for both binary and multi-class problems. However, the IRs so far considered (up to IR = 500), while substantial, and well beyond those typically addressed in other works, are even so not as high as might be encountered in certain specialist applications, such as cheminformatics [134]. For such extreme scenarios, with IRs up to 1000s, it would be worthwhile to consider other potential avenues to explore. One such approach, to generalise the form of the activation function for a network's output layer in order to ease the difficulty of classification of the minority type, will be the topic of the next chapter, where it will be shown that this new approach can be used effectively both alongside, and separately from, a classification-focused loss.

Chapter 5

Boosting Performance for Extreme IR: ASTra, a Complementary Approach to Imbalanced Classification

While the results of the previous chapter were highly encouraging, it is possible in extreme scenarios (IRs in the 1000s) that classification-focused loss functions alone might not be enough. An alternative approach to handling high IRs would be to look at making the classification of the minority class intrinsically easier. This can be done, in the binary case, by modifying the form of the activation function in a neural network's output layer. This chapter will motivate, describe, and demonstrate a specific modification for binary classification known as ASTra (Asymmetric Sigmoid Transfer function). ASTra will be used both alone, with a conventional binary cross-entropy (BCE) loss function, and in combination with the G-Mean loss function of Chapter 3. The G-Mean is chosen over more the advanced loss function variants introduced in Chapter 4 because it had already, in Chapter 3, been demonstrated to

be workable and advantageous for both the UCI and Pump & Dump datasets and is, therefore, (arguably) the simplest choice.

The proposed approach is tested on datasets with IRs from 588.24 to 4000 which, in particular, contain very few minority examples (in some datasets, as few as five). Problems with such extreme levels of class imbalance are not in fact uncommon, and arise in application areas already discussed such as fraud detection [135] and cheminformatics [134]. The results for these test datasets, using neural networks with 2 to 12 hidden units, are demonstrated to be comparable to, or better than, equivalent results obtained in a recent study that examined a wide range of complex, hybrid data-level ensemble classifiers [8]. It will be seen that ASTra alone is a powerful intervention for high-IR problems, though the results of this chapter contain some evidence that at even higher levels of IR it could be combination of ASTra with a classification-based loss function that would prove optimal. In addition, this chapter also presents a new means of monitoring training and validation performance, especially valuable in cases of high class imbalance, that could potentially be used in any training regime, independently of the proposed methods.

5.1 Origin and Derivation of the ASTra Transform

The origin of the ASTra transform lies in fact entirely outside of machine learning, in the fields of botany and ecology. The ASTra transform is based on the integrated form of the Richards Growth Equation [136]. The Richards Growth Equation has been used, for example, to model vegetation growth and the growth of young mammals and birds, and is given by

$$\frac{\partial y}{\partial t} = ay[1 - \frac{y}{K}^{b}], \tag{5.1}$$

in which t represents time, a is a constant equal to the relative growth rate at very low values of y, K is the maximum observable value of y, and b is a constant that allows the shape of the sigmoid curve to be varied (for $b > 1$ the maximum slope of the curve is when $y > K/2$, for $b = 1$ it is when $y = K/2$ and when $b < 1$ it is when $y < K/2$).

The appeal of the Richards equation in the current context is that its integrated form is an asymmetric sigmoid curve whose maximum gradient is at a variable position τ within the function's output range. This point of maximum slope can in machine learning can be identified with the activation function's threshold parameter. This is a more substantial modification of the concept of threshold than models which simply apply an offset within the conventional sigmoid.

The constant a within the original form of the Richards equation is here set to $a = 1$, since outside of a biological context there is no reason to treat the lower asymptote of the 'growth curve' differently to the upper one. Furthermore, since the range of an activation function should be $\in [0, 1]$ if it is to define a probability, it is obvious to take $K = 1$. Now also replacing the input variable t which originally represented time, to x, the Richards equation becomes

$$\frac{\partial y}{\partial x} = y[1 - y^b]. \tag{5.2}$$

With the further assumption that $b > 0$ (since a negative decision threshold would not make sense), the integrated form of the above equation becomes

$$y = (1 + e^{(d-bx)})^{\frac{-1}{b}}, \tag{5.3}$$

where d is a parameter that indirectly defines the value of x at which $y = 0.5$. For $d = 0$ and $b = 1$, the integrated Richards equation plainly recovers the regular

sigmoid function. The maximum slope, obtained at $b(1+b)^{-\frac{1+b}{b}}$, for $b=1$ is $2^{-2}=\frac{1}{4}$. The value of y corresponding to the maximum slope is $(1+b)^{-\frac{1}{b}}$ (for $b=1$, $y=2^{-1}=1/2$, and the value of x corresponding to the maximum slope is $\frac{d-\ln b}{b}$.

From this point, there are two possible ways to go. If the value of x that corresponds to the maximum slope (threshold) is not of concern, then d can be set as $d=0$, and therefore

$$y(x)=(1+e^{-bx})^{(-1/b)}. \tag{5.4}$$

However, if the value of x for the maximum slope is desired to be when $x=0$, then it is required that $d=\ln b$ and therefore that

$$y(x)=(1+be^{-bx})^{(-1/b)}. \tag{5.5}$$

In either case the implied threshold is

$$\tau=(1+b)^{(-1/b)}. \tag{5.6}$$

(It should be noted that the threshold equation cannot be algebraically inverted to give $b(\tau)$; if this is needed, e.g. for initial threshold setting, a numerical solver is required). In the end, the second choice, Equation 5.5, was selected because it generated a form that most closely resembled the original sigmoid. The only difference was that the maximum slope occurred at a y value greater or lesser than 0.5, as shown in Figure 5.1.

Turning now to potential constraints on the value of the parameter b, it quickly became apparent that

$$f(x,b) = (1 + be^{-bx})^{(-1/b)} \tag{5.7}$$

did not give a useful result for $0 < b < 1$; only values of $b \geq 1$ (where $b = 1$ recovers the original sigmoid) provide the desired range of shapes as shown in Figure 5.1.

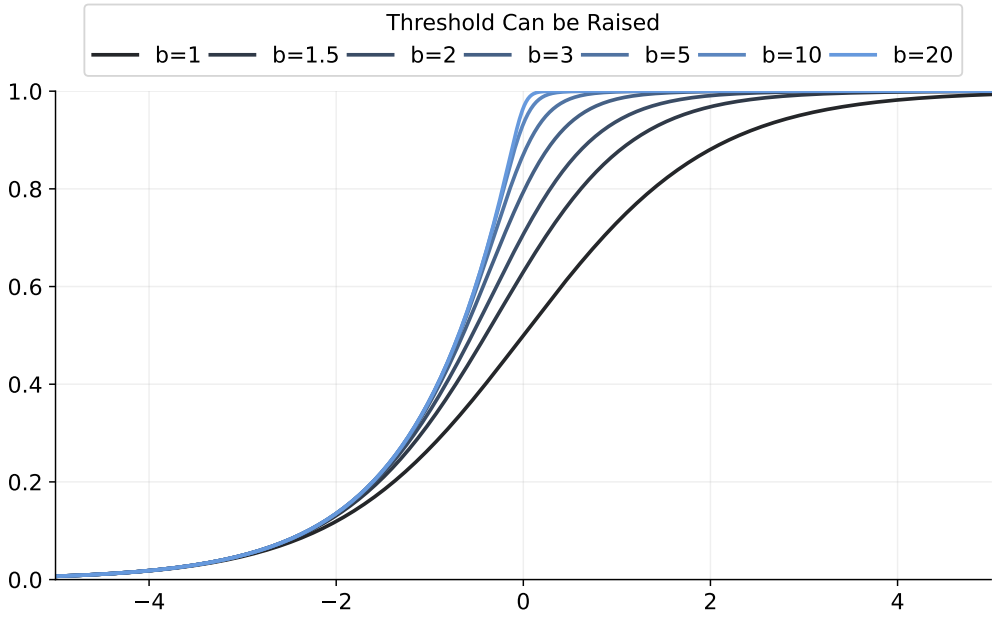

Figure 5.1: Activation profile for $f(x,b) = (1 + be^{-bx})^{(-1/b)}$ for different values of b. At $b = 1$ this is equivalent to the standard sigmoid function. However as b increases, the threshold (i.e. point of maximum slope) *increases*. This setup, the one derived most directly from the Richards equation, works effectively for minority-0 labels but for a conventional minority-1 requires adjustment.

Figure 5.1 also shows that, as b increases, the threshold increases. This is actually the opposite of what is desired for the standard situation where the minority class label is equal to 1 and the majority equal to 0 (also known as the minority-1 setup), since the idea would then be to *lower* the threshold to reduce the number of false negatives. This output transform could in principle still be used for a conventional minority-1 scenario if the class labels were flipped. However, given that the ability to retain conventional class labels might be an important consideration for the user, it was felt an extension should be found that is compatible with the

more common setup of minority-1 labels. Therefore, $f(x,b)$ was adapted so that the threshold can now be lowered to achieve easier classification of the minority, via a simple geometric transformation: $g(x,b) = 1 - f(-x,b)$. Either setup can be employed, depending on what way round labels are applied (minority-1 vs. minority-0), and whether the key objective is to reduce *false negatives*, (the more common task), or *false positives*. However, care should be taken in the changes required, these being highlighted in Table 5.1: although the equations for $z(y,\tau)$, $\frac{\partial}{\partial y}z(y,t)$ and $\frac{\partial}{\partial \tau}z(y,\tau)$ (the need for, and form of, the z-transform is explained later in subsection 5.2) remain the same, everything else requires an important change.

Table 5.1: Summary of ASTra equations based on target setup. The ASTra formulation varies depending on whether targets are labelled as minority-0 or minority-1. $z(y,\tau)$, $\frac{\partial}{\partial y}z(y,t)$ and $\frac{\partial}{\partial \tau}z(y,\tau)$ remain the same for both.

	Label Convention	
Function	Minority-0	Minority-1
$y(x,b)$	$(1+be^{-bx})^{-\frac{1}{b}}$	$1-(1+be^{bx})^{-\frac{1}{b}}$
$\frac{\partial}{\partial x}y(x,b)$	$y(1-y^b)$	$(1-y)(1-(1-y)^b)$
$\tau(b)$	$(1+b)^{(-1/b)}$	$1-(1+b)^{(-1/b)}$
$z(y,\tau)$	$\frac{y(1-\tau)}{y(1-\tau)+(1-y)\tau}$	
$\frac{\partial}{\partial y}z(y,\tau)$	$\frac{z^2\tau}{y^2(1-\tau)}$	
$\frac{\partial}{\partial b}y(x,b)$	$-\frac{y}{b}[\log y + \frac{1}{b}(1-y^b)(1-xb)]$	$\frac{(1-y)}{b}[\log(1-y)+$ $\frac{1}{b}(1-(1-y)^b)(1+xb)]$
$\frac{\partial}{\partial \tau}z(y,\tau)$	$-\frac{z^2(1-y)}{y(1-\tau)^2}$	
$\frac{\partial \tau}{\partial b}$	$-\frac{\tau}{b}[\log\tau+\tau^b]$	$\frac{\tau}{b}[\log\tau+\tau^b]$

The activation profile for a minority-1 setup is shown in Figure 5.2, with further examples (this time labelled by threshold τ, for $\tau = 0.5, 0.25, 0.05$, rather than parameter b, and showing both (a) the activation function, and (b) its derivative) in Figure 5.3. As a sidenote, it has so far been found that 0.05 is the lowest workable

value for τ (attempts to compress all majority examples into $< 5\%$ of the output range having led to numerical instabilities), and that therefore no values of τ less than 0.05 will be used in the experiments to follow.

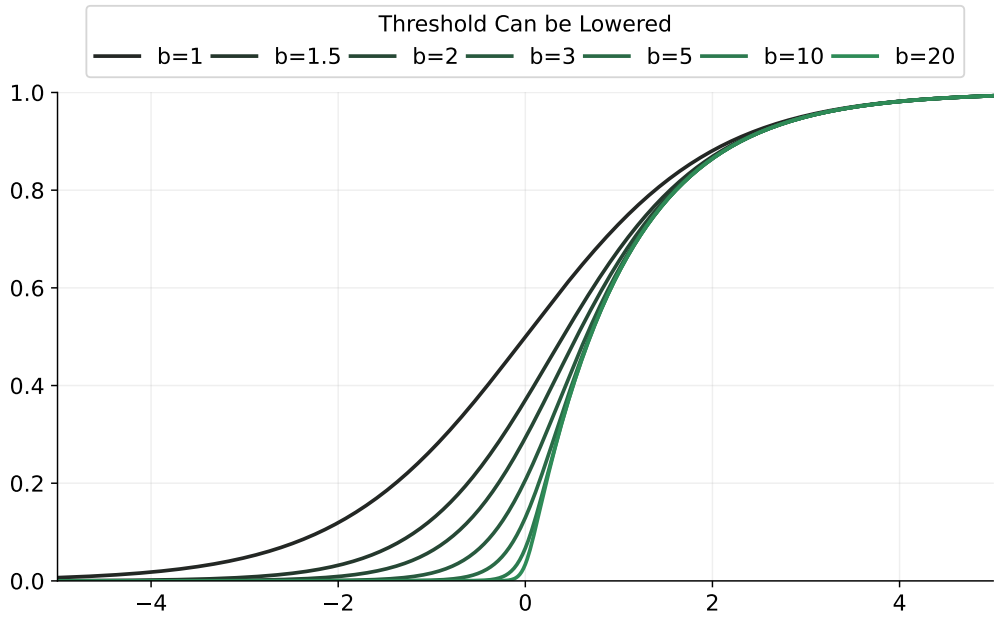

Figure 5.2: Activation profile for $g(x,b) = 1 - f(-x,b)$ for different values of b. At $b = 1$ this is equivalent to the standard sigmoid function. In this case, as b increases, the threshold (i.e. point of maximum slope) *decreases*. This setup is best for reducing false negatives in the conventional minority-1 setup.

5.2 Adaptation of Loss Functions to Accommodate ASTra

At first sight, this looks easy: simply replace the usual sigmoid output, in the selected loss function, by (assuming, from hereon, a conventional minority-1 setup)

$$ASTra(x,b) = 1 - f(-x,b) = 1 - (1 + be^{bx})^{(-1/b)} \tag{5.8}$$

However, it is not always so straightforward, as an examination of ASTra's effect of the computation of a binary cross-entropy (BCE) loss will show.

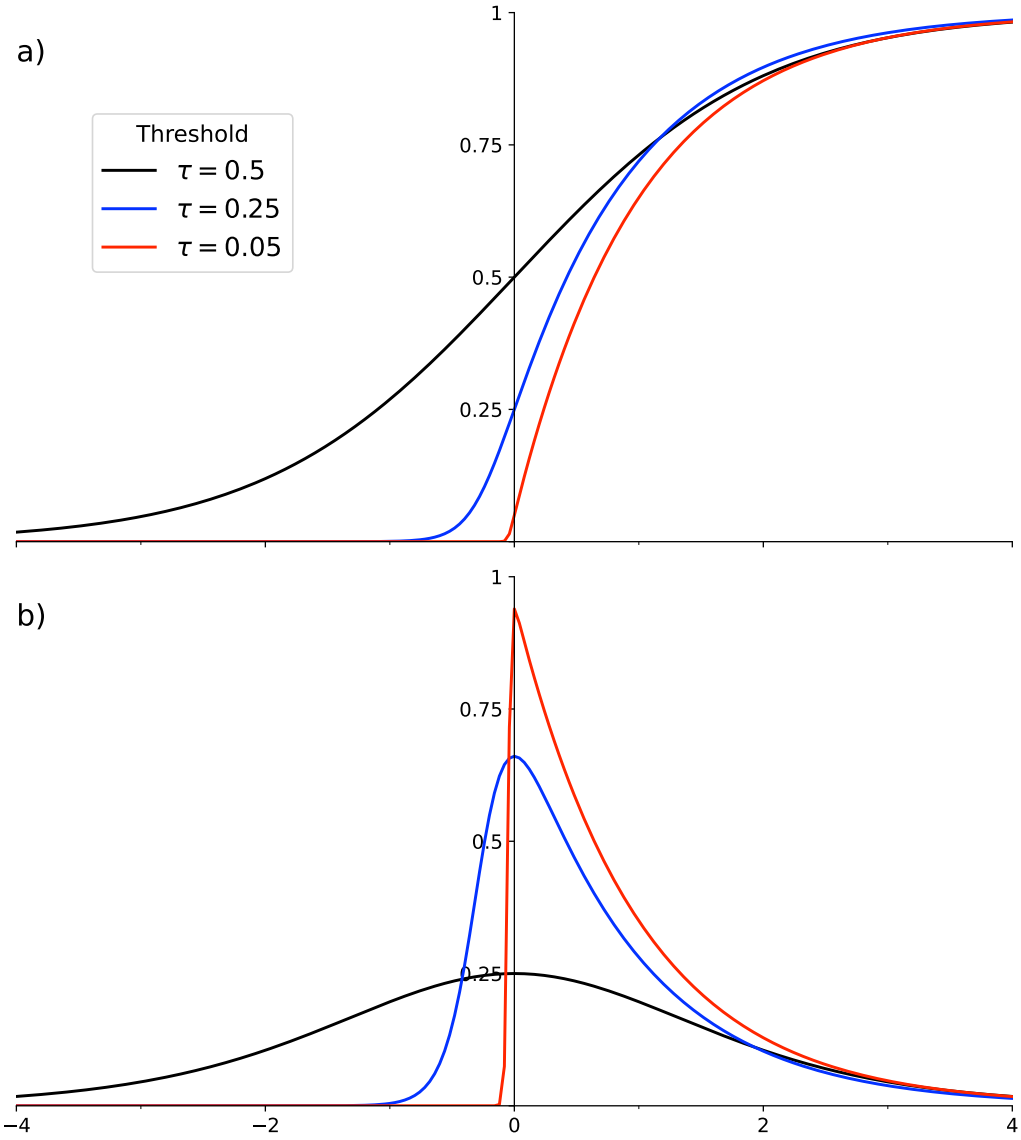

Figure 5.3: (a) Activation function $ASTra(x,b)$ and (b) its first derivative, for $\tau(b) = 0.5, 0.25, 0.05$.

Consider the contribution $J_{BCE}(\hat{y},y) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$ of a single example, with prediction $\hat{y} = ASTra(x,b)$ and target $y \in \{0,1\}$, to the train set loss. As is apparent in Figure 5.4 a), for $b = 1$ ($\tau = 0.5$), for any input $x > 0.0$, $J_{BCE}(\hat{y},0) > J_{BCE}(\hat{y},1)$, correctly, since $\hat{y}$ will be on the 'correct side' of the threshold and so closer to target 1 than to 0, with the converse also correctly being true for inputs $x < 0.0$. However, for $b > 1$ this is not true for all values of the input x; there will be a range of x, specifically, $0 < x < (\log[(2^b - 1)/b])/b$, within which the loss contributions are wrongly ordered. Fig. 2(b) shows an example of this for $\tau = 0.25$ ($b = 7.396$); within the shaded band it is the case that $J_{BCE}(\hat{y},0) < J_{BCE}(\hat{y},1)$, so outputs which are 'more right than wrong' are instead being informed that the opposite is the case.

The fundamental problem is that currently $J_{BCE}(\hat{y},0)$ and $J_{BCE}(\hat{y},1)$ cross at 0.5, not at τ, and the solution is to use within the BCE loss not $\hat{y}$ but a transform $z(\hat{y},\tau)$, given by

$$z(\hat{y},\tau) = \frac{\hat{y}(1-\tau)}{\hat{y}(1-\tau) + (1-\hat{y})\tau}, \tag{5.9}$$

such that z, $1-z$ will now cross at τ, as can be seen in Figure 5.5, for $\tau = 0.5, 0.25$, and 0.05, and hence the loss contributions $J_{BCE}(z,0)$ and $J_{BCE}(z,1)$ will do so, also. For some loss functions, such as mean squared error, it is unnecessary to transform $\hat{y}$ in this way. However, the z-transform should be carried out for any classification loss function that implicitly pivots around a threshold of 0.5, including all loss functions, such as GMN, derived from the confusion matrix.

5.3 Learning the Threshold Value

It would in principle be possible to use a fixed value of τ and set this via hyperparameter optimisation. However, in experiments so far, it has appeared that a fixed

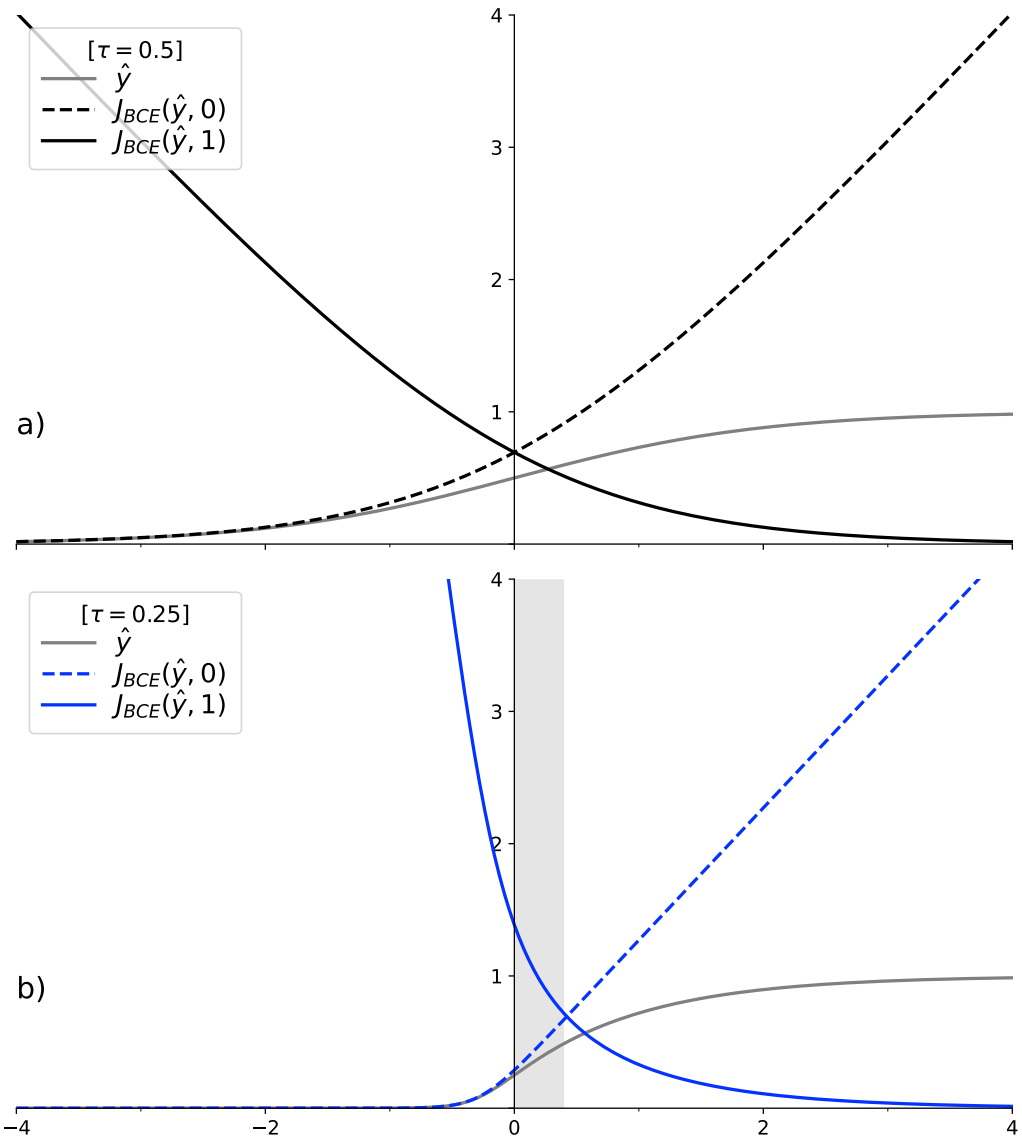

Figure 5.4: Loss functions for targets 0 and 1 for (a) $\tau(b) = 0.5$ and (b) $\tau(b) = 0.25$

$\tau < 0.5$ during weights-learning is non-optimal, and that it is better to begin with a somewhat larger threshold value that then decreases during the network's learning process. There is necessarily therefore a question of how τ should be decreased. While it would be possible to experiment with a schedule of decrease, for example, exponential or linear, a more attractive option is to learn the value of the underlying 'slope' parameter, b, alongside the network's weights.

For a conventional minority target of 1, the relevant values of b should be

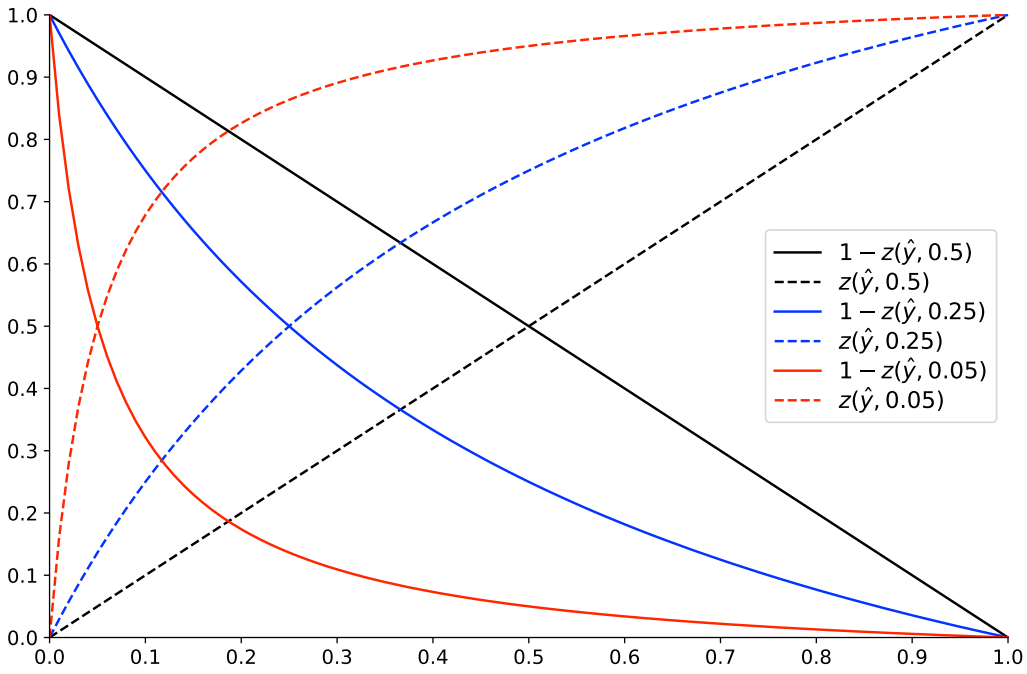

Figure 5.5: z-transforms for $\tau = 0.5, 0.25, 0.05$

constrained to be ≥ 1(corresponding to a threshold $\tau \leq 0.5$). Given this, it is useful to express b in terms of an unconstrained underlying parameter β:

$$b = \begin{cases} 2+\beta & \beta > 0 \\ 1+e^{\beta} & \text{otherwise} \end{cases} . \tag{5.10}$$

It is preferable to begin with $b(\beta)$ in its linear range, i.e., $b \geq 2$, so that the threshold can change sufficiently quickly at the start of the learning process. The value $\tau_{init} = 0.25$ ($b = 7.396$) is used here, around halfway between the achievable threshold maximum of 0.5 and the recommended (on stability grounds) minimum of 0.05, but clearly this is a value that could be treated as a hyperparameter and optimised.

5.4 Data, Experimental Process, and Performance Measurement

5.4.1 Data

There are not many easily available datasets with the high level of imbalance required for this work. However, a recent study [8] has looked at data with IRs ranging from 9.08 to 970.6, with data made available online [137]. It was also useful that these data were used in [8] to compare HD-Ensemble, a high-performing hybrid, data-level ensemble classifier devised by that study's authors, to seven other state-of-the-art ensemble classifiers, including RUSBoost [86]. The three most imbalanced datasets from [8], named there as *skinnonskin*, *cod-rna*, and *ijcnn1* (IR = 970.6) were downloaded from [137].[1] In addition to the original files, the two largest datasets from [137] were minority-undersampled to create more challenging problems (IRs of 3500, 4000, with one positive example in each of five folds), extracting five positive samples randomly at each run, to eliminate bias created by 'good' or 'bad' subset picks.

Table 5.2: Datasets used in this work. The shaded bars correspond to the original datasets of [8] and the unshaded bars to our additional minority-undersampled datasets

Dataset	**IR**	**Abb.**	n_x	m_{tot}	m_1
skinnonskin	588.24	*skin-588*	3	20034	34
above, undersampled	4000	*skin-4000*	3	20005	5
cod-rna	763.27	*cod-763*	8	19871	26
above, undersampled	3500	*cod-3500*	8	17505	5
ijcnn1	970.60	*ijcnn-971*	22	4858	5

The properties of the datasets used here are summarised in Table 5.2, in which n_x is the number of input features, m_{tot} the total number of examples in the dataset,

[1] Modified, where necessary, to record target-1 for the minority class and target-0 for the majority, a requirement for the application of the ASTra activation function.

and m_1 the number of minority-positive examples. It should be noted that while a similar process to [8] was followed in creating stratified test folds, 10×5CV testing rather than 5×10CV was used, using three folds for training, and one each for validation and testing, as it was unclear how tenfold stratified sampling had been done in [8] in the case of *ijcnn1*, with only five positive examples.

5.4.2 Details of the Experimental Process

The focus of this work is not the test problems of Table 5.2, per se, but a comparison of methodologies, and therefore simple choices were made for network architectures, with no attempt to optimise learning rates, etc., in order to get the best solution in each case. For each test problem, four candidates were compared: training with a binary cross-entropy loss (referred to as BCE); training with a binary cross-entropy loss and ASTra transform (BCE-ASTra); training with the GMN loss function (GMN); and training with the GMN loss function and ASTra transform (GMN-ASTra). The architecture for each test problem consisted of a neural network with one hidden layer of Leaky ReLU neurons (having a negative slope coefficient of 0.3) and a single ASTra output (which can be converted to the standard sigmoid when $b = 1$). The number of hidden units n_h, was determined using the formula $n_h = ceil((n_x + n_y)/2)$, where n_x is the number of input features and n_y denotes the number of output units (in this case, consistently set to 1 to represent the single ASTra output). Network weights used He initialisation in the hidden layer and Glorot initialisation in the output layer. All models were trained using Adam, with a training rate of $\eta = 0.001$ for the network weights. η_b, the training rate associated with the slope parameter b, was set according to the adaptive rule of Equation 5.11,

$$\eta_b = \begin{cases} \min(k_{mult} \cdot \eta_b, \eta_{bmax}) & \text{if } e\text{-}ratio > 1.0 \\ \max(k_{dec} \cdot \eta_b, \eta_{bmin}) & \text{if } e\text{-}ratio < 1.0 \\ \eta_b & \text{otherwise} \end{cases}, \tag{5.11}$$

the objective of which was to allow the threshold adjustment to proceed more quickly when the model is struggling to handle its train set class imbalance, but slow down when conditions are less taxing, as measured by the *e-ratio* of Section 5.4.3. Here, η_{bmin} (also the starting value of η_b) is chosen to be 0.01, η_{bmax}, $k_{mult} = 1.1$, and $k_{dec} = 0.99$. These values were not optimised, in line with a focus on comparison of training methods, rather than on outright performance.

The one form of problem-specific optimisation performed was the extraction of the best weights (and b value, where relevant) with respect to a validation set. However, it was unfeasible to use early stopping as best validation performance could in some cases occur late, after an early setback. Therefore, training was conducted for a fixed 10,000 epochs, extracting a 'best-on-val' set of parameters, for whatever epoch this occurred, with respect to the chosen validation performance measure (FNR_{apx}, for the reasons discussed in the following section).

5.4.3 Performance Measurement

Two aspects of performance measurement need to be considered, that which is carried out during the training process, and that which is used to assess the results on the test sets. For the latter, the G-Mean is used, as it is the basis of one of our considered loss functions, and one of the two performance measures quoted in [8]. However, AUROC is not included, as in [8], as this metric has been criticised in the case of imbalanced data [53], and we found it to distinguish very poorly between the methods of this paper. Instead, MCC is substituted as the second measure, due to its

high regard as a performance metric for imbalanced data.

Turning to in-training performance monitoring, another innovation is introduced in that we retain the explanatory value, for this purpose, of confusion matrix based metrics, but exchange the usual, counting-measure forms for ones based on the approximated matrix—i.e., we use standard formulae for G-Mean, etc., but base the calculation on the matrix elements of the approximated confusion matrix rather than confusion matrix. Metrics derived from CM_{apx} re-introduce the 'by how much?' that counting-measure classification metrics lose. This is particularly important when monitoring validation performance, for which, in this work, we select to monitor FNR_{apx}, the approximated false negative rate, rather than MCC_{apx}, for example, as in these extreme scenarios it is test set FNR that largely determines classification (in terms of MCC, G-Mean) success. The other element of in-training performance monitoring relates to the train set itself, rather than to validation. We record

$$e\text{-}ratio = \frac{FNR_{apx}}{FPR_{apx}} \tag{5.12}$$

for the train set as a way to measure the difficulty a model has with the train set class imbalance; this ratio is used in Equation 5.11 but also of intrinsic interest in understanding how the methods deal with class imbalance, with a large difference being discovered between BCE loss and GMN loss training, as will be seen later.

5.5 Results

The results of the 10×5CV study are summarised in Table 5.3, below, in which averages and standard deviations (in brackets) are quoted for both G-Mean and MCC, and where BCE, GMN refer to the loss functions (J_{BCE}, J_{GMN}) used.

The large standard deviations, apparent also in the results of [8], are a consequence of fold-based aggregation of results in a situation where a small number of

Table 5.3: Mean fold-aggregated test performance (10×5CV), with standard deviations bracketed. Cells with bold type and shading denote winners or ties, where to be deemed a winner a method needs to outperform its competitors with $p \leq 0.05$.

	Without ASTra		With ASTra	
	BCE	**GMN**	**BCE**	**GMN**
skin-588				
G-Mean	**0.892 (0.273)**	**0.966 (0.051)**	**0.981 (0.041)**	**0.946 (0.126)**
MCC	**0.892 (0.273)**	**0.966 (0.051)**	**0.976 (0.042)**	0.763 (0.273)
skin-4000				
G-Mean	**0.760 (0.436)**	**0.879 (0.331)**	**0.800 (0.408)**	**0.840 (0.374)**
MCC	0.760 (0.436)	**0.843 (0.368)**	**0.800 (0.408)**	0.688 (0.382)
cod-763				
G-Mean	0.449 (0.396)	0.768 (0.206)	**0.862 (0.208)**	**0.843 (0.206)**
MCC	0.433 (0.391)	0.419 (0.301)	**0.760 (0.229)**	0.542 (0.291)
cod-3500				
G-Mean	0.240 (0.436)	0.357 (0.486)	**0.640 (0.490)**	**0.519 (0.509)**
MCC	0.240 (0.436)	0.255 (0.397)	**0.421 (0.369)**	0.206 (0.225)
ijcnn-971				
G-Mean	0.000 (0.000)	0.120 (0.331)	**0.478 (0.508)**	**0.596 (0.497)**
MCC	-0.000 (0.001)	0.072 (0.208)	**0.162 (0.179)**	**0.179 (0.162)**

false negatives (in the case of the *skin-4000*, *cod-3500*, and *ijcnn-971* datasets, a single false negative) on an individual fold can cause its G-Mean or MCC to drop from near-unity to zero. This in turn causes some results, with a sample size of 50 (10CV), to be not statistically significantly distinguishable; e.g., in the case of the *skin-588* dataset, the G-Means of the four methods cannot be separated at $p \leq 0.05$ in spite of an apparent-win by BCE with ASTra (BCE-ASTra). Nevertheless, despite the necessity of a registering a proportion of the results as 'ties', a pattern emerges, with the problems forming a hierarchy of difficulty, at each level of which different methods will be appropriate:

- *skin-588* is an example of a problem for which no additional measures to address IR need to be taken, since vanilla BCE training can do as well (subject to the above note on statistical significance testing) as any of the more advanced

methods.

- *skin-4000*, the minority-undersampled version of *skin-588*, is a more difficult problem, for which BCE training falls behind with respect to MCC. For this problem, adopting the principle that a minimal intervention is to be preferred, GMN training, without the ASTra transform, would be recommended.

- *cod-763*, *cod-3500*, and *ijcnn-971* are examples of problems that appear to clearly benefit from the ASTra transform. For the *cod* variants it is BCE-ASTra that does best with respect to both G-Mean and MCC, having fewer false positives for these problems, while for *ijcnn-971* BCE-ASTra and GMN-ASTra tie.

The above list does not include a problem for which GMN-ASTra does best. However, we would expect such problems to exist. GMN-ASTra attacks false negatives at the expense of a somewhat increased number of false positives, and if it is possible to avoid this by using BCE-ASTra, that is desirable. But it is likely there are problems that require an increase in false positives to be tolerated in order to have any chance of recognising instances of the minority class (i.e., of having G-Means and MCCs above zero), and in such cases we would expect GMN-ASTra to be preferred.

For the three datasets we have in common with [8] it is possible to draw performance comparisons with that work, though retaining our concerns about the use of AUROC for highly imbalanced data, which we found did not distinguish well between the methods. The two best-performing methods of [8], in relation to G-Mean and AUROC, were the HD-Ensemble method of [8] and RUSBoost [86]. For *skin-588*, GMN, BCE-ASTra, and GMN-ASTra all had statistically significantly indistinguishable performance from RUSBoost, with respect to both metrics. For

cod-763, the two ASTra variants tied with RUSBoost on AUROC, but beat RUSBoost on G-Mean, while on *ijcnn-971* they achieved a better G-Mean than state-of-the-art HD-Ensemble. Given especially that we used small, single-layer neural networks, with that used for *skin-588* having only three neurons, we consider these results encouraging.

As a means of illuminating the difference between our four considered methods, Figure 5.6 shows the *e-ratio* (as defined in Equation 5.12), for the *skin-4000* (most highly imbalanced) problem. The GMN variants find handling the imbalance much easier, with *e-ratio* values up to five orders of magnitude lower; also, while ASTra is taxing in terms of *e-ratio*, this can be offset by the use of GMN. While extreme scenarios will likely require ASTra, this figure is indicative as to why, for some imbalanced problems, GMN alone would be sufficient.

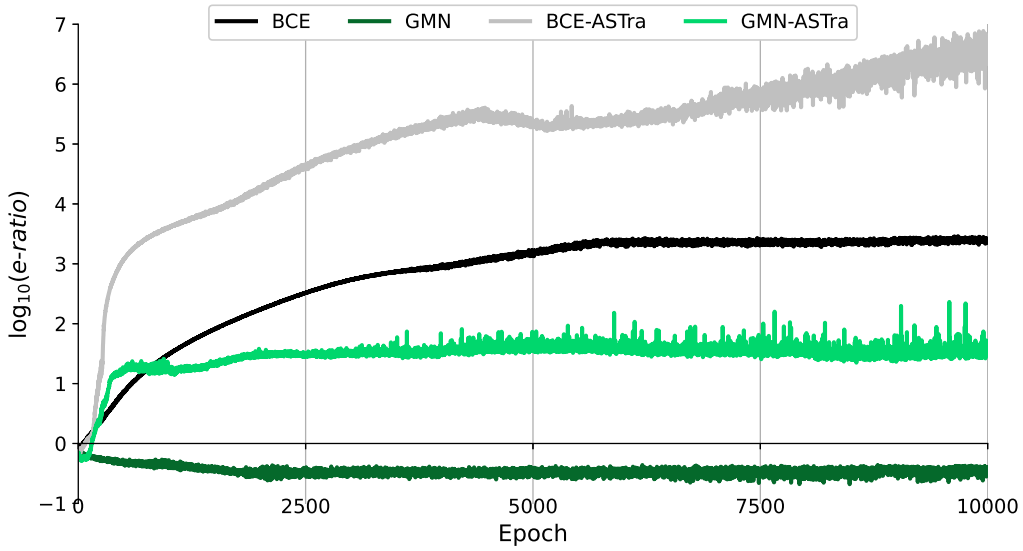

Figure 5.6: Fold-averaged training behaviour for *skin-4000*, in terms of the (logged) *e-ratio* for each considered method.

5.6 Discussion

This chapter proposed a novel output layer activation function, named ASTra (Asymmetric Sigmoid Transfer function), which makes the classification of minority examples in scenarios of high imbalance more tractable. While this new activation can be used on its own, it can also be combined with any of the classification-focused loss functions introduced in the previous chapters.

The proposed methods were applied to the most imbalanced datasets in a recent extensive study of ensemble classifiers [8] and achieved performances comparable to those reported in [8] for RUSBoost [86], despite our restricting this initial investigation to the use of neural networks with 3–12 neurons in a single hidden layer. It should be emphasised, however, that for complex problems it is essential to explore the space of model architectures, since even in the case of a high IR the main difficulties may in fact lie in an insufficient model architecture. It is also recommended that a progressive exploration of the methods is conducted, beginning with the GMN loss alone, then advancing to the addition of the ASTra output layer activation function, as needed.

It was notable that in a number of instance ASTra with the BCE loss function did better in terms of MCC than both the G-Mean loss and the G-Mean loss with ASTra. Does this mean that the combination of the G-Mean (or other classification-based loss function) with ASTra is not, after all, likely to be effective for very high IRs? The answer to this question is found in a consideration of two topics. First, the contribution of, and importance of, false positives, and second the the definition of “high” - which an IR of 3000 is certainly extremely high in comparison to the IRs handled in the current ML literature, IRs in some fields, such as fraud detection [135], potentially even higher.

BCE training, precisely because it is less biased, tends to handle false positives

better than G-Mean training. A higher IR also means a higher potential for false positives simply because there are so many more negative examples, and this can depress measures like the MCC substantially even if all of the minority class are categorised perfectly. While false positives tend to be less important in their implications than false negatives, they are at least a nuisance and so should be eliminated if possible - provided this can be done without creating more false negatives. This is what ASTra appears to be able to do, for most of the levels of IR so far considered. However, for even higher IRs, it is likely that BCE training, even augmented by ASTra, will begin to fail on the minority class. In such instances, it would be expected that a combination of ASTra with a classification-based loss, such as the MCC loss, could prove optimal, and experiments with even higher IRs will therefore be a component in future work.

Moving on to other potential avenues of exploration, it is not currently obvious how to extend ASTra to the multi-class case. It should be noted that it would not in fact be easy to deduce the form of a conventional sigmoid from examining the form of softmax for a two-class problem - for one thing, the output layer for two-class softmax has two times the number of weight parameters that the sigmoid would have, and thus an element of redundancy in its representation. The extension of ASTra to the multi-class case, which will be the subject of future work, is likely to require embedding some of the key ASTra concepts in a framework that does not rely on the idea of a neuron threshold function, and is thus likely to go considerably beyond ASTra's original inspiration in the Richards Growth Equation.

Finally, a novel use of metrics based on an approximated confusion matrix for performance monitoring during training was introduced. This is of potential interest entirely outside of frameworks introduced in this thesis, as it relates to the more effective monitoring of training performance for any classification problem,

addressed using any algorithm that would give rise to a confusion matrix.

Chapter 6

Summary & Conclusions

The work of this thesis focused on algorithm-level approaches to the difficulties caused by class imbalance in classification problems. As explained in Chapter 2, algorithm-level approaches are one category of solution proposed for class imbalance, the others being data-level manipulation of the underlying datasets, and hybrid approaches which combine multiple techniques or use ensembling approaches. Algorithm-level approaches have the benefit of not requiring manipulations of the data that may introduce bias or error, and the specific algorithm-level approaches proposed in this thesis have the additional advantage of being easy to implement, not requiring a machine learning applications developer to adopt an entirely new category of model with many parameters to be set or learned. In this final chapter, the contributions of this thesis are discussed, future work is then proposed, and concluding remarks finally made.

6.1 Discussion of Contributions

In Chapter 3, the concept of an approximated confusion matrix was introduced. This matrix is a continuous representation of a confusion matrix, which can be constructed from model predictions and true class labels of any batch-based backpropagation training approach. Using this approximated confusion matrix, it was demonstrated

that differentiable approximations of commonly used (non-differentiable) classification scoring metrics, such as the G-Mean, were possible, allowing their use as loss function in training. This form of algorithm-level approach satisfies the aim of developing a simple and easily adoptable methodology for handling class imbalance, as it requires only that a user switch from a conventional loss function to one such as the G-Mean; no changes to the model itself are required. This general approach, that of deriving loss functions from an approximated confusion matrix, was introduced in this chapter via the example of the approximated G-mean (GMN) loss. The G-Mean is a balanced metric of true positive rate and true negative rate, and was chosen for these first experiments as its approximated form has a simple derivative. Two experiments were performed to demonstrate the value of the new GMN loss. Firstly, its outperformance over traditional losses (mean squared error and binary cross entropy) was demonstrated on a set of class-imbalanced UCI datasets. Secondly, a more challenging dataset, 'pump and dump' was explored, with the GMN loss remaining highly competitive when compared to standard loss approaches augmented with popular data-level interventions (random undersampling and SMOTE).

In Chapter 4, the work of the previous chapter was extended in two ways, while also moving into deep learning application domain. In the first contribution of this chapter, two novel confusion matrix based losses, the F1 loss, based on the approximated F1-Score, and the MCC loss, based on the approximated Matthews Correlation Coefficient, were introduced, and these losses, together with the GMN loss from Chapter 3 were applied to two image segmentation datasets. Results from this experiment showed their relative benefit compared with the use of a standard loss (binary cross entropy) and two recently used domain-specific losses, the Dice loss and Focal loss. This experiment demonstrated first that a range of novel loss functions can be constructed, and second that such loss functions can be used to

address challenging problems requiring deep learning networks, in this way going beyond the work of Chapter 3, which required only the use of relatively shallow multilayer perceptron networks. The second contribution of this chapter was to demonstrate that novel loss functions derived from an approximated confusion matrix can be extended to a multi-class problems. Multi-class macro-averaged versions of the GMN, F1 and MCC loss were constructed and used as loss functions for the second (classifier) learning stage of a two-stage approach to a common multi-class class-imbalanced task known as the Long Tailed Recognition (LTR) problem. State-of-the-art results were achieved on overall accuracy, with the GMN, F1, and MCC losses outperforming all recent work, and representing a considerable advance, with the predominant advantage being seen, as might be expected, for the least frequent classes.

In Chapter 5, another, complementary algorithm-level approach to imbalanced classification was introduced. A new output layer activation function, named ASTra (Asymmetric Sigmoid Transfer function), was proposed, which can make the classification of minority examples in scenarios of high class imbalance more tractable. ASTra can be used with any loss objective, but for extremely challenging cases, it appears best to combine it with an imbalance-robust loss such as the G-Mean from Chapter 3. ASTra was tested on datasets with imbalance ratios from 588.24 to 4000 and with very few minority examples (in some datasets, as few as five). Results using neural networks with from 2 to 12 hidden units were demonstrated to be comparable to, or better than, equivalent results obtained in a recent study that deployed a wide range of complex hybrid data-level classifiers. In addition to this, it was noted that a novel use of metrics based on the approximated confusion matrix was possible, namely their use for performance monitoring during training, independently of the choice of loss function, in order to ensure that apparent improvements in classifica-

tion performance (in terms of a reduction, for example, in cross entropy loss) were genuine, i.e., were not merely the result of a progressively improving performance on the majority class alone.

6.2 Future Work

A number of areas for future work are apparent:

- The newly proposed methods could be applied to a larger variety of data sets and class imbalance levels, in both binary and multi-class scenarios, including experiments with data having even higher IRs than those seen in Chapter 5.

- While the novel loss functions were extended to a multi-class framework, this has not yet been done for the ASTra activation function, and this would appear to be a potentially fruitful area for further work.

- Some authors [138] have suggested feature selection may be important in addressing class imbalance; this could be investigated, as a potential further complement to the approaches of this thesis.

- The work of this thesis has applied the proposed methods only in a neural network context. Novel differentiable losses could also be integrated with gradient-boosted decision tree ensembles (e.g. XGBoost [139]), which are more commonly used for tabular datasets, and are considered easier to work with (since they are less sensitive to initial conditions).

- The use of more complicated loss functions means a higher likelihood of complex and non-convex training loss landscapes, with a multiplicity of local and global minima, with different global minima yielding models with different generalization abilities. Thus, it may be very valuable to combine novel loss functions with the SAM (Sharpness Aware Minimisation) framework [140];

SAM is a method to smooth loss landscapes in the vicinity of minima, and in this way improve generalisation, and is a technique that has shown recent strong uptake and a lot of promise across a variety of benchmark datasets (e.g., CIFAR-10, CIFAR-100, ImageNet).

- The use of regularisation is an increasingly strong theme in work with class-imbalanced datasets. The state of the art results of Chapter 4 incorporated regularisation in the first stage of the two-stage training process (that part that used a conventional cross entropy loss). However, a very recent work [141] has proposed a novel regularisation method for class imbalanced problems for which a more complex loss function has been chosen. This method has not yet been used in combination with any of the novel loss functions of this thesis, hence would clearly be a valuable avenue of work to pursue.

- Deep learning networks (DNNs) suggest a number of areas of future work, and experiments with deep learning methods for addressing class imbalance in the context of big data and class rarity will prove valuable to the future of big data analytics. It has been commented that deep learning research exploring the effect of class imbalance on non-image data is limited and should be expanded upon [15] ; in the context of this thesis, which has looked only at CNNs for image problems, more work would be required with non-convolutional DNNs to determine if the methods presented in the thesis will generalise well to alternative DNN architectures, e.g. RNNs.

- Current research on imbalanced data recognises that class imbalance is aggravated by other data-intrinsic characteristics, among which class overlap stands out as one of the most harmful [142]. Further experiments that allow the manipulation of both IR and the level of class overlap could address the

degree to which the methods of this thesis are compatible with the presence of high overlap as well as high IR.

- While data manipulations such as oversampling are in general best avoided, there may be some datasets, with very few examples of the minority class, for which it would be favourable to combine advanced data oversampling techniques (e.g. GANs, DeepSMOTE [68]), with the novel losses and/or ASTra transform proposed here.

6.3 Concluding Remarks

This thesis has demonstrated that relatively simple interventions - the use of a classification-focused loss function, and the generalisation of the sigmoid output layer activation function to an asymmetric forms with an adaptive threshold - can be very effective means to address the class imbalance problem, competitive with far more complex methodologies, and in some cases achieving results that outstrip the state of the art. However, as noted in the list of possible topics of future work, there are a number of places in which the proposed novel loss function and/or the ASTra function could potentially be combined with other methodologies, both simple and complex, to achieve even better results. It would arguably be of great value to develop a generic toolkit for imbalanced data, allowing the combination of multiple approaches as needed, and on the basis of the results presented here it would appear that the proposed methods might be a valuable addition to such a toolkit.

Appendix A

Second-Order Derivatives

All the work in this thesis uses first order gradient based optimisation for training classifier models. In theory, however, second order optimisation methods such as Newton, Gauss-Newton and Quasi-Newton approaches could also be used with the approximated loss functions such as the G-Mean, but they would require estimation of second order partial derivatives.

Given the approximated confusion matrix

$$CM_{apx} = \begin{pmatrix} \sum_i^n (1-\hat{y}_i)\cdot(1-y_i) & \sum_i^n \hat{y}_i \cdot (1-y_i) \\ \sum_i^n (1-\hat{y}_i)\cdot y_i & \sum_i^n \hat{y}_i \cdot y_i \end{pmatrix}, \tag{A.1}$$

in which $\hat{y}_i$ and y_i are the predicted and target values for the ith example, then the approximated square of the G-mean is given by

$$GMN^2_{apx} = \frac{1}{m_0 \cdot m_1}[TN_{apx} \cdot TP_{apx}], \tag{A.2}$$

where m_0 and m_1 are the numbers of negative and positive examples, respectively and the approximated true negatives and true positives are given by:

$$
\begin{aligned}
TN_{apx} &= \sum_i (1-\hat{y}_i)\cdot(1-y_i), \\
TP_{apx} &= \sum_i \hat{y}_i \cdot y_i.
\end{aligned}
\tag{A.3}
$$

From Chapter 3 that its first partial derivative with respect to the network output $\hat{y}$ for input pattern q is:

$$
\frac{\delta GMN^2_{apx}}{\delta \hat{y}_q} = \frac{1}{m_0 \cdot m_1}[(\frac{\partial}{\partial \hat{y}_q} TN_{apx}) \cdot TP_{apx} + TN_{apx} \cdot (\frac{\partial}{\partial \hat{y}_q} TP_{apx})]. \tag{A.4}
$$

If it is assumed that it is known how to calculate

$$
\frac{\partial \hat{y}_q}{\partial w}, \tag{A.5}
$$

for some parameter w that is relevant to the model's output $\hat{y}$, then for pattern q:

$$
\begin{aligned}
\frac{\partial GMN^2_{apx}}{\partial w} &= \frac{\partial \hat{y}_q}{\partial w} \cdot \frac{\partial GMN^2_{apx}}{\partial \hat{y}_q} \\
&= \frac{1}{m_0 \cdot m_1} \frac{\partial \hat{y}_q}{\partial w}[-(1-y_q)\cdot TP_{apx} + TN_{apx} \cdot (1-y_q)]
\end{aligned}
\tag{A.6}
$$

Assuming it is also known how to calculate

$$
\frac{\partial^2 \hat{y}_q}{\partial v \partial w}, \tag{A.7}
$$

for any two output-relevant parameters v and w, the second partial derivative (of the approximated G-Mean squared GMN^2) with respect to these parameters is:

$$\frac{\partial^2 GMN_{apx}^2}{\partial v \partial w} = \frac{1}{m_0 \cdot m_1}\left[\frac{\partial^2 \hat{y}_q}{\partial v \partial w}[-(1-y_q)\cdot TP_{apx} + TN_{apx}\cdot y_q] \right.$$
$$\left. -2\frac{\partial \hat{y}_q}{\partial v}\cdot\frac{\partial \hat{y}_q}{\partial w}[(1-y_q)\cdot y_q]\right]$$

Bibliography

[1] Daniel W. Hook, Simon J. Porter, and Christian Herzog. Dimensions: Building context for search and evaluation. *Frontiers in Research Metrics and Analytics*, 3:23, 2018. https://www.frontiersin.org/articles/10.3389/frma.2018.00023/pdf.

[2] Wikipedia. Confusion matrix — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Confusion%20matrix&oldid=1088688630`, 2022. Accessed 20-June-2022.

[3] Hmrishav Bandyopadhyay. Image segmentation: Deep learning vs traditional. `https://www.v7labs.com/blog/image-segmentation-guide`, May 2022. Accessed: 2022-06-20.

[4] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

[5] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. *arXiv preprint arXiv:2203.14197*, 2022.

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, pages 9268–9277, 2019.

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[8] Zhi Chen, Jiang Duan, Li Kang, and Guoping Qiu. A hybrid data-level ensemble to enable learning from highly imbalanced dataset. *Information Sciences*, 554:157–176, 2021.

[9] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.

[10] Jannes Münchmeyer, Jack Woollam, Andreas Rietbrock, Frederik Tilmann, Dietrich Lange, Thomas Bornstein, Tobias Diehl, Carlo Giunchi, Florian Haslinger, Dario Jozinović, et al. Which picker fits my data? a quantitative evaluation of deep learning based seismic pickers. *Journal of Geophysical Research: Solid Earth*, 127(1):e2021JB023499, 2022.

[11] Kanika and Jimmy Singla. Class balancing methods for fraud detection using deep learning. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pages 395–400. IEEE, 2022.

[12] Yutaka Sasaki. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.

[13] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[14] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition, 2016.

[15] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[16] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2):195–215, 1998.

[17] Ronald Pearson, Gregory Goney, and James Shwaber. Imbalanced clustering for microarray time-series. In *Proceedings of the ICML*, volume 3, 2003.

[18] Neelam Rout, Debahuti Mishra, and Manas Kumar Mallick. Handling imbalanced data: a survey. In *International proceedings on advances in soft computing, intelligent systems and applications*, pages 431–443. Springer, 2018.

[19] Kevin S Woods, Christopher C Doss, Kevin W Bowyer, Jeffrey L Solka, Carey E Priebe, and W Philip Kegelmeyer Jr. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(06):1417–1436, 1993.

[20] Sivaramakrishnan Rajaraman, Prasanth Ganesan, and Sameer Antani. Deep learning model calibration for improving performance in class-imbalanced medical image classification tasks. *PloS one*, 17(1):e0262838, 2022.

[21] Elias Tappeiner, Martin Welk, and Rainer Schubert. Tackling the class im-

balance problem of deep learning based head and neck organ segmentation. *arXiv preprint arXiv:2201.01636*, 2022.

[22] Kitsuchart Pasupa, Supawit Vatathanavaro, and Suchat Tungjitnob. Convolutional neural networks based focal loss for class imbalance problem: A case study of canine red blood cells morphology classification. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–17, 2020.

[23] Anju Jain, Saroj Ratnoo, and Dinesh Kumar. Addressing class imbalance problem in medical diagnosis: a genetic algorithm approach. In *2017 international conference on information, communication, instrumentation and control (ICICIC)*, pages 1–8. IEEE, 2017.

[24] Zhenchuan Li, Mian Huang, Guanjun Liu, and Changjun Jiang. A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection. *Expert Systems with Applications*, 175:114750, 2021.

[25] Matias Di Martino, Federico Decia, Juan Molinelli, and Alicia Fernández. Improving electric fraud detection using class imbalance strategies. In *ICPRAM (2)*, pages 135–141, 2012.

[26] Sara Makki, Zainab Assaghir, Yehia Taher, Rafiqul Haque, Mohand-Said Hacid, and Hassan Zeineddine. An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, 7:93010–93022, 2019.

[27] David A Cieslak, Nitesh V Chawla, and Aaron Striegel. Combating imbalance in network intrusion datasets. In *GrC*, pages 732–737. Citeseer, 2006.

[28] Joffrey L Leevy, Taghi M Khoshgoftaar, and Jared M Peterson. Mitigating class imbalance for iot network intrusion detection: a survey. In *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 143–148. IEEE, 2021.

[29] Taehyung Lee, Ki Bum Lee, and Chang Ouk Kim. Performance of machine learning algorithms for class-imbalanced process fault detection problems. *IEEE Transactions on Semiconductor Manufacturing*, 29(4):436–445, 2016.

[30] Georg Helbing and Matthias Ritter. Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189–198, 2018.

[31] S Mostafa Mousavi, William L Ellsworth, Weiqiang Zhu, Lindsay Y Chuang, and Gregory C Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications*, 11(1):1–12, 2020.

[32] Weiqiang Zhu and Gregory C Beroza. Phasenet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019.

[33] Minh Thanh Vo, Anh H Vo, Trang Nguyen, Rohit Sharma, and Tuong Le. Dealing with the class imbalance problem in the detection of fake job descriptions. *Computers, Materials & Continua*, 68(1):521–535, 2021.

[34] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[35] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[36] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[37] Gary M Weiss and Foster Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of artificial intelligence research*, 19:315–354, 2003.

[38] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56, pages 111–117. Citeseer, 2000.

[39] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[40] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Mining data with rare events: a case study. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 132–139. IEEE, 2007.

[41] Gary M Weiss. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19, 2004.

[42] Ioannis Markoulidakis, Ioannis Rallis, Ioannis Georgoulas, George Kopsiaftis, Anastasios Doulamis, and Nikolaos Doulamis. Multiclass confusion matrix reduction method and its application on net promoter score classification problem. *Technologies*, 9(4):81, 2021.

[43] Josephine Akosa. Predictive accuracy: A misleading performance measure for highly imbalanced data. In *Proceedings of the SAS Global Forum*, volume 12, 2017.

[44] Hongyu Guo and Herna L Viktor. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM Sigkdd Explorations Newsletter*, 6(1):30–39, 2004.

[45] Marcus A Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II*, volume 2, pages 2–1, 2003.

[46] Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern recognition*, 40(12):3358–3378, 2007.

[47] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

[48] Harald Cramér. Mathematical methods of statistics (pms-9), volume 9. In *Mathematical Methods of Statistics (PMS-9), Volume 9*. Princeton university press, 2016.

[49] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[50] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[51] Vincent Van Asch. Macro-and micro-averaged evaluation measures. *Belgium: CLiPS*, 49, 2013.

[52] Peter Flach and Meelis Kull. Precision-recall-gain curves: Pr analysis done right. *Advances in neural information processing systems*, 28, 2015.

[53] Cheng G Weng and Josiah Poon. A new evaluation measure for imbalanced datasets. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*, pages 27–32, Glenelg, South Australia, 2008.

[54] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[55] Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. A survey on graphical methods for classification predictive performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 23(11):1601–1618, 2011.

[56] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[57] Niall M Adams and David J Hand. Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition*, 32(7):1139–1147, 1999.

[58] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[59] Muhammad Atif Tahir, Josef Kittler, Krystian Mikolajczyk, and Fei Yan. A multiple expert approach to the class imbalance problem using inverse random under sampling. In *International workshop on multiple classifier systems*, pages 82–91. Springer, 2009.

[60] Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pages 4368–4374. IEEE, 2016.

[61] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[62] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[63] José A Sáez, Julián Luengo, Jerzy Stefanowski, and Francisco Herrera. Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.

[64] Venkata Pavan Kumar Turlapati and Manas Ranjan Prusty. Outlier-smote: A refined oversampling technique for improved detection of covid-19. *Intelligence-based medicine*, 3:100023, 2020.

[65] Josey Mathew, Ming Luo, Chee Khiang Pang, and Hian Leng Chan. Kernel-based smote for svm classification of imbalanced datasets. In *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pages 001127–001132. IEEE, 2015.

[66] Benjamin X Wang and Nathalie Japkowicz. Imbalanced data set learning with synthetic samples. In *Proc. IRIS machine learning workshop*, volume 19, page 435, 2004.

[67] Tiantong Wang, Daniel Trugman, and Youzuo Lin. Seismogen: Seismic waveform synthesis using gan with application to seismic data augmentation. *Journal of Geophysical Research: Solid Earth*, 126(4):e2020JB020077, 2021.

[68] Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[69] Zhiqiang Wan, Yazhou Zhang, and Haibo He. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[70] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.

[71] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239, 2017.

[72] Joan Pastor-Pellicer, Francisco Zamora-Martínez, Salvador España-Boquera, and María José Castro-Bleda. F-measure as the error function to train neural networks. In *International Work-Conference on Artificial Neural Networks*, pages 376–384. Springer, 2013.

[73] David Twomey and Denise Gorse. A neural network cost function for highly class-imbalanced data sets. In *ESANN 2018-Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 207–212. i6doc. com, 2018.

[74] Chunkai Zhang, Guoquan Wang, Ying Zhou, and Jiayao Jiang. A new approach for imbalanced data classification based on minimize loss learning. In *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pages 82–87. IEEE, 2017.

[75] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[76] Lluís-F Hurtado, José-Angel González, and Ferran Pla. Choosing the right loss function for multi-label emotion classification. *Journal of Intelligent & Fuzzy Systems*, 36(5):4697–4708, 2019.

[77] Mariusz Paradowski, Michał Spytkowski, and Halina Kwaśnicka. A new f-score gradient-based training rule for the linear model. *Pattern Analysis and Applications*, 22(2):537–548, 2019.

[78] Aaron Scherzinger, Philipp Hugenroth, Marike Rüder, Sven Bogdan, and Xiaoyi Jiang. Multi-class cell segmentation using cnns with f1-measure loss function. In *German Conference on Pattern Recognition*, pages 434–446. Springer, 2018.

[79] Kumar Abhishek and Ghassan Hamarneh. Matthews correlation coefficient loss for deep convolutional networks: Application to skin lesion segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 225–229. IEEE, 2021.

[80] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8577–8584, 2019.

[81] Michael D Richard and Richard P Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991.

[82] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[83] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1367–1381, 2018.

[84] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[85] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2781–2794, 2019.

[86] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

[87] Mikel Galar, Alberto Fernández, Edurne Barrenechea, and Francisco Herrera. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern recognition*, 46(12):3460–3471, 2013.

[88] Zhiquan Qi, Bo Wang, Yingjie Tian, and Peng Zhang. When ensemble learning meets deep learning: a new deep support vector machine for classification. *Knowledge-Based Systems*, 107:54–60, 2016.

[89] Reza Rasti, Mohammad Teshnehlab, and Son Lam Phung. Breast cancer diagnosis in dce-mri using mixture ensemble of convolutional neural networks. *Pattern Recognition*, 72:381–390, 2017.

[90] Grigory Antipov, Sid-Ahmed Berrani, and Jean-Luc Dugelay. Minimalistic cnn-based ensemble model for gender prediction from face images. *Pattern recognition letters*, 70:59–65, 2016.

[91] Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pages 199–208. PMLR, 2015.

[92] Dheeru Dua and Casey Graff. UCI machine learning repository. `http://archive.ics.uci.edu/ml/index.php`, 2017. Accessed: 2022-06-20.

[93] Roberto Battiti. Accelerated backpropagation learning: Two optimization methods. *Complex systems*, 3(4):331–342, 1989.

[94] Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency {Pump-and-Dump} scheme. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1609–1625, 2019.

[95] Shuo Wang, Leandro L Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. *IEEE transactions on neural networks and learning systems*, 29(10):4802–4821, 2018.

[96] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.

[97] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[98] William R Crum, Oscar Camara, and Derek LG Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE transactions on medical imaging*, 25(11):1451–1461, 2006.

[99] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[100] Stanislav Nikolov, Sam Blackwell, Alexei Zverovitch, Ruheena Mendes, Michelle Livne, Jeffrey De Fauw, Yojan Patel, Clemens Meyer, Harry Askham, Bernardino Romera-Paredes, et al. Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy. *arXiv preprint arXiv:1809.04430*, 2018.

[101] L Soler, A Hostettler, V Agnus, A Charnoz, J Fasquel, J Moreau, A Osswald, M Bouhadjar, and J Marescaux. 3d image reconstruction for comparison of algorithm database: A patient specific anatomical and medical image database. *IRCAD, Strasbourg, France, Tech. Rep*, 2010.

[102] Michael P Snyder, Long Cai, Jay Shendure, Cole Trapnell, Shin Lin, Dana Jackson, Guo-Cheng Yuan, Qian Zhu, and Ruben Dries. The human body at cellular resolution: the nih human biomolecular atlas program. *Nature*, 574(7777):187–192, 2019.

[103] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on*

Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

[104] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[105] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126(9):961–972, 2018.

[106] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[107] Jan Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5-6):367–374, 2004.

[108] RTH. Introduction to $R_K coefficient, 2021. Last accessed 11 April 2022.$

[109] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2021.

[110] Ingrid C Romero, Shu Kong, Charless C Fowlkes, Carlos Jaramillo, Michael A Urban, Francisca Oboh-Ikuenobe, Carlos D'Apolito, and Surangi W Pun-

yasena. Improving the taxonomy of fossil pollen using convolutional neural networks and superresolution microscopy. *Proceedings of the National Academy of Sciences*, 117(45):28496–28505, 2020.

[111] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017.

[112] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[113] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[114] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020.

[115] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[116] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object

recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

[117] Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9495–9504, 2021.

[118] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[119] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–350, 2018.

[120] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

[121] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020.

[122] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021.

[123] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network con-

struction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

[124] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[125] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[126] Cao Yue, M Long, J Wang, Zhu Han, and Q Wen. Deep quantization network for efficient image retrieval. In *Proc. 13th AAAI Conf. Artif. Intell.*, pages 3457–3463, 2016.

[127] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[128] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[129] John E Moody. Note on generalization, regularization and architecture selection in nonlinear learning systems. In *Neural Networks for Signal Processing Proceedings of the 1991 IEEE Workshop*, pages 1–10. IEEE, 1991.

[130] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. cvpr. 2016. *arXiv preprint arXiv:1512.03385*, 2016.

[131] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

[132] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021.

[133] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 235–244, 2021.

[134] Wojciech Marian Czarnecki and Krzysztof Rataj. Compounds activity prediction in large imbalanced datasets with substructural relations fingerprint and eem. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 2, pages 192–197. IEEE, 2015.

[135] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475, 2013.

[136] FJ Richards. A flexible growth function for empirical use. *Journal of experimental Botany*, 10(2):290–301, 1959.

[137] Zhi Chen, Jiang Duan, Li Kang, and Guoping Qiu. HD-Ensemble datasets, 2021. Last accessed 11 April 2022.

[138] Guang-Hui Fu, Yuan-Jiao Wu, Min-Jie Zong, and Lun-Zhao Yi. Feature selection and classification by minimizing overlap degree for class-imbalanced data in metabolomics. *Chemometrics and Intelligent Laboratory Systems*, 196:103906, 2020.

[139] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[140] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[141] Bosung Kim, Youngjoong Ko, and Jungyun Seo. Novel regularization method for the class imbalance problem. *Expert Systems with Applications*, 188:115974, 2022.

[142] Miriam Seoane Santos, Pedro Henriques Abreu, Nathalie Japkowicz, Alberto Fernández, Carlos Soares, Szymon Wilk, and João Santos. On the joint-effect of class imbalance and overlap: a critical review. *Artificial Intelligence Review*, pages 1–69, 2022.