


November 2021

## **Uncertainty Quantification in Deep and Statistical Learning with applications in Bio-Medical Image Analysis**

K. Ruwani M. Fernando  
*University of South Florida*

Follow this and additional works at: <https://digitalcommons.usf.edu/etd>

 Part of the [Statistics and Probability Commons](#)

---

### **Scholar Commons Citation**

Fernando, K. Ruwani M., "Uncertainty Quantification in Deep and Statistical Learning with applications in Bio-Medical Image Analysis" (2021). *USF Tampa Graduate Theses and Dissertations*.  
<https://digitalcommons.usf.edu/etd/9673>

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact [scholarcommons@usf.edu](mailto:scholarcommons@usf.edu).

Uncertainty Quantification in Deep and Statistical Learning with applications in Bio-Medical  
Image Analysis

by

K. Ruwani M. Fernando

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Mathematics & Statistics  
College of Arts and Sciences  
University of South Florida

Major Professor: Chris P. Tsokos, Ph.D.  
Kandethody M. Ramachandran, Ph.D.  
Lu Lu, Ph.D.  
Yicheng Tu, Ph.D

Date of Approval:  
November 18, 2021

Keywords: Class Imbalance, Uncertainty Estimation, Bayesian Deep Learning, High Dimensional  
Data, Biomedical Imaging

Copyright © 2021, K. Ruwani M. Fernando

## DEDICATION

*Dedicated to my beloved family  
who have always been  
my strength and encouragement.*

## ACKNOWLEDGMENT

I would first like to express my profound gratitude to my major advisor, Professor Chris P. Tsokos for his unequivocal guidance, enthusiastic encouragement, and useful critiques of this research work. Thank you Prof. Tsokos for being a great supervisor.

My grateful thanks are extended to Dr. Kandethody Ramachandran, Dr. Lu Lu, and Dr. Yicheng Tu for serving in my supervisory committee of the Ph.D. research and for being very supportive throughout my Ph.D. program. I would also like to thank Dr. Andrei Barbos for chairing my Ph.D. defense session.

I would like to extend my appreciation to the Florida Center for Cybersecurity at University of South Florida and Citi group, Tampa, for the summer internship opportunities. My internship experiences helped me to choose the right direction and successfully complete my dissertation.

Finally, a heartiest thank to my dear family for their support and encouragement throughout my study. Thank You for believing in me and motivating me to do my best.

## TABLE OF CONTENTS

List of Tables .....	iv
List of Figures .....	v
Abstract .....	vii
Chapter 1 Introduction .....	1
1.1 Overview .....	1
1.2 General Objectives .....	2
1.2.1 Class Imbalance Learning and Confidence Calibration .....	2
1.2.2 Uncertainty Quantification in Deep Learning .....	2
1.2.3 Sparse Bayesian Time-to-Event Modeling and Radiomics .....	3
1.3 Contributions .....	3
1.4 Dissertation Structure .....	4
1.5 Notation .....	4
Chapter 2 Deep learning: Preliminaries .....	6
2.1 Mathematical and Statistical Foundations in Deep Learning .....	6
2.1.1 Feed-forward Neural Networks .....	7
2.1.2 Convolutional Neural Networks .....	8
2.2 Probabilistic Deep Learning .....	9
2.2.1 The Bayesian Interpretation of Neural Network Learning .....	9
2.2.2 Variational Inference .....	12
2.2.3 Bayesian Convolutional Neural Networks .....	13
Chapter 3 Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks <sup>1</sup> .....	15
3.1 Introduction .....	15
3.2 Related Work .....	17
3.2.1 Class Imbalance .....	17
3.2.2 Confidence Calibration .....	19
3.3 Dynamically Weighted Balanced (DWB) Loss .....	19
3.3.1 Loss Function Formulation .....	19
3.3.2 Improving Calibration using DWB loss .....	22
3.3.3 DWB Loss Function Gradients .....	25
3.4 Experiments .....	26
3.4.1 Experimental set-up and Evaluation .....	26
3.4.2 Experiment 1: Cyber Intrusion Detection .....	27
3.4.2.1 Dataset Description .....	28

3.4.2.2	Implementation (Intrusion Detection System Model Overview) ...	28
3.4.2.3	Experimental Results .....	31
3.4.3	Experiment 2: Skin Lesion Diagnosis .....	32
3.4.3.1	Dataset Description .....	33
3.4.3.2	Implementation Details .....	33
3.4.3.3	Experimental Results .....	34
3.5	Contributions and Concluding Remarks .....	37
Chapter 4	Bayesian-based Probabilistic Deep learning for Uncertainty Quantification with applications in Bio-medical Image Segmentation .....	38
4.1	Introduction .....	38
4.2	Related Work .....	40
4.3	Segmentation model construction from a probabilistic perspective .....	41
4.3.1	Parameter Uncertainty and Probabilistic layers .....	41
4.3.2	Data-dependent Uncertainty .....	44
4.3.3	Uncertainty Estimation .....	45
4.4	Experiments and Results .....	46
4.4.1	3D MRI Brain Tumor segmentation .....	46
4.4.2	Quantitative Evaluation .....	48
4.4.3	Experimental Results .....	49
4.5	Contributions and Concluding Remarks: .....	52
Chapter 5	Radiomics in Neuro-oncology: A Sparse Bayesian approach for modeling high-dimensional data in Glioblastoma survival prediction .....	53
5.1	Introduction .....	53
5.2	Related Work .....	54
5.3	Background, concepts and notation .....	55
5.3.1	Survival analysis preliminaries .....	55
5.3.2	Time to Event Regression: Accelerated Failure Time (AFT) model ...	56
5.4	Sparse Bayesian Survival Regression .....	57
5.4.1	Bayesian AFT model formulation .....	57
5.4.2	Prior Specification .....	58
5.4.3	Model fitting and Posterior Inference .....	58
5.5	Application to survival prediction of patients with Glioblastoma .....	59
5.5.1	Experimental Set-up .....	59
5.5.2	Exploratory Analysis .....	60
5.5.3	Radiomics analysis for survival time prediction .....	62
5.5.3.1	Quantitative Feature Extraction .....	64
5.5.3.2	Statistical Analysis and model building .....	64
5.5.4	Experimental Results .....	67
5.6	Contributions and Concluding Remarks .....	70
Chapter 6	Conclusions and Future Research .....	71
References	.....	73
Appendix A	Description of Radiomic Features .....	86

Appendix B Copyright Permission ..... 87

## LIST OF TABLES

Table 3.1:	The distribution of network flows in each attack category .....	29
Table 3.2:	CICIDS2017 Dataset: Average metric values (Percentages).....	32
Table 3.3:	CICIDS2017 Dataset: Class-wise Classification Performance .....	32
Table 3.4:	CICIDS2017 Dataset: Calibration Performance.....	32
Table 3.5:	The distribution of skin lesion diagnostic category .....	34
Table 3.6:	ISIC2019 Dataset: Average Metric Values (Percentages).....	35
Table 3.7:	ISIC2019 Dataset: class-wise classification (Percentages) .....	36
Table 3.8:	ISIC2019 Dataset: Calibration Metrics .....	36
Table 4.1:	Segmentation Performance Evaluation: Results on BRATS2020 training and validation sets as generated by the online portal .....	50
Table 4.2:	Uncertainty Estimation Performance Evaluation: Results on BRATS2020 training and validation sets as generated by the online portal.....	51
Table 5.1:	Statistical information for the training set in BRATS2020 survival data.....	59
Table 5.2:	Survival Function Transformations.....	65
Table 5.3:	Bayesian variable selection and their summary statistics. ....	69
Table 5.4:	Glioblastoma survival prediction performance evaluation on BRATS2020 data set .....	69
Table A.1:	Description of Radiomic Features Extracted from Pyradiomics .....	86



## LIST OF FIGURES

Figure 2.1:	Convolution and Max pooling layer operations in a CNN architecture.....	9
Figure 2.2:	Deterministic and Probabilistic neural networks. ....	10
Figure 2.3:	An example of a BCNN. ....	14
Figure 3.1:	Comparisons among proposed Dynamically Weighted Balanced (DWB) Loss and other commonly used losses for classification. ....	22
Figure 3.2:	Visualization of the weight term based on predicted probability of ground truth class ( $p$ ) on long-tailed CICIDS2017 data. ....	23
Figure 3.3:	Network activity flow distribution with network flow-count varying sharply across different attack categories. ....	29
Figure 3.4:	1D-CNN Model Architecture.....	31
Figure 3.5:	A sample of different skin lesion categories from the ISIC2019 data set.....	33
Figure 3.6:	Skin Lesion Diagnosis distribution with lesion count varying sharply across different diagnosis categories. ....	34
Figure 3.7:	The architecture of EfficientNet-B0. ....	35
Figure 3.8:	Schematic diagram of the dual-input neural network model architecture composed of a 2D-CNN (EfficientNet-B3) and fully connected model. ....	35
Figure 3.9:	Probability calibration plot for ISIC2019 data. ....	36
Figure 4.1:	An example of glioblastoma brain tumor in T1, T1-contrast, T2 and FLAIR modalities and ground truth segmentation.....	39
Figure 4.2:	Epistemic and Aleatoric uncertainty.....	40
Figure 4.3:	Process overview for developing a deep probabilistic neural network via variational inference. ....	43
Figure 4.4:	Schematic U-Net like network architecture of the proposed probabilistic model.....	47

Figure 4.5:	Qualitative results of four representative subjects on the BRATS 2020 data set. ....	50
Figure 4.6:	Evaluation of the mean segmentation performance of the probabilistic SegNet and comparison against deterministic U-Net and MC dropout modeling approaches on validation data. ....	51
Figure 5.1:	Kaplan-Meier plots of overall survival in all cases by resection status. ....	61
Figure 5.2:	Fitted probability density and survival functions with Gamma distribution for the entire cohort of patients with Glioblastoma BRATS2020 data set. ....	62
Figure 5.3:	Overview of radiomics workflow for Survival Prediction. ....	63
Figure 5.4:	An example of a fused image of 4 modalities in Glioblastoma BRATS2020 data set. ....	64
Figure 5.5:	The cluster-map map provides a clear visual representation of feature correlations. ....	65
Figure 5.6:	Plots of transformed survival functions for Log-Normal distribution, Log-logistic Distribution and Weibull distribution. ....	66
Figure 5.7:	Bayesian linear regression model with only age of the patient as a predictor. ...	66
Figure 5.8:	Graphical model depicting the conditional dependencies. ....	67
Figure 5.9:	Trace plots for parameters indicating convergence. ....	68
Figure B.1:	E-mail received from copyright clearance center (e-mail attachment is in Figure B.2.) ....	87
Figure B.2:	Permission grant statement. ....	88

## ABSTRACT

Deep Learning (DL) has achieved the state-of-the-art performance across a broad spectrum of tasks. From a statistical standpoint, deep neural networks can be construed as universal function approximators. Although statistical modeling and deep learning methods are well-established as independent areas of research, hybridization of the two paradigms via probabilistic deep networks is an emerging trend. Through development of novel analytical methods under the statistical and deep-learning framework, we address some of the major challenges encountered in the design of intelligent systems which include class imbalance learning, probability calibration, uncertainty quantification and high dimensionality. When modeling rare events, existing methodologies require re-sampling strategies or algorithmic modifications. On the contrary, we introduce a cost sensitive approach that could be promptly applied to any deep neural network architecture. Our research corroborates that the proposed approach leads to significant performance gains in highly imbalanced data and results in improved calibration. Moreover, deterministic neural nets are ignorant to the uncertainty associated with their predictions and tend to produce overconfident predictions, resulting in unreliable model predictions. Uncertainty-aware deep networks provide additional insights to model predictions and produce a more informed decision and thus, is indispensable in applications where the acceptable margin of error is significantly low. To this end, we present a Bayesian-based deep probabilistic learning approach that provides a principled framework for handling uncertainty. Furthermore, we address high dimensionality in time-to-event modeling which is a common problem in computational biology such as in genomics. Our results suggest that in the presence of limited but high dimensional data, inducing sparsity through shrinkage priors under the Bayesian framework is a potent alternative to the machine learning methods. With theoretical justification and sound empirical validation on data across different domains of cyber-security and healthcare we provide validity for the proposed methods.

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

Deep Learning (DL) models with different deep architectures and learning paradigms have made breakthroughs in various application domains. From a statistical point of view, neural networks are non-linear function approximators that excel at learning complex feature representations from data. Engrafting statistical methodologies in deep networks can be a more effective alternative in the design of automated systems than methods that rely merely on one discipline. For instance, in healthcare, integrated statistical and deep learning methods have recently emerged as a new direction in the automation of the medical practice unifying multi-disciplinary knowledge in medicine, statistics, and artificial intelligence. This dissertation demonstrates how to formulate deep and statistical learning-based models to address many core issues in the design of intelligent systems: class imbalance, calibration, uncertainty, and high-dimensionality.

While the contributions made here have wide applicability in different domains, we mainly focus on Bio-medical imaging. Over the past decade, the capacities of medical imaging devices have increased tremendously and consequently, new statistical and deep learning methods for medical imaging analysis emerged. Clinical imaging has progressed into a powerful diagnostic tool enabling identification of morphologic and biological changes facilitating histopathological diagnosis of diseases, treatment, and monitoring therapeutic response. For instance, in neuro-oncology, brain tumor segmentation in medical scans aids in the evaluation of structural abnormalities and identification of tumor-related complications. For a comprehensive overview on biomedical image segmentation algorithms linking the two disciplines of statistics and deep learning, the interested reader is referred to [1]. Similarly, dermoscopic images in skin lesion diagnosis provide indispensable information for the development of clinical models in skin cancer. Our experimental results suggest that model-driven classical statistics and data-driven deep learning is a potent combination for developing automated systems in clinical oncology.

## 1.2 General Objectives

The current study is concentrated on developing analytical and algorithmic solutions in machine learning and statistical analysis for solving pressing issues in the design of automated systems, more specifically, class imbalance learning, uncertainty quantification and high dimensionality in time-to-event analysis. Contributions made in each chapter is briefly discussed below.

### 1.2.1 Class Imbalance Learning and Confidence Calibration

Imbalanced class distribution is an inherent problem in many real-world classification tasks where the minority class is the class of interest. Many conventional statistical and machine learning classification algorithms are subject to frequency bias and learning discriminating boundaries between the minority and majority classes could be challenging. To address the class distribution imbalance in deep learning, in chapter 3 (published in [2]) we propose a class re-balancing strategy based on a class-balanced dynamically weighted loss function where weights are assigned based on class frequency and predicted probability of ground truth class. The ability of dynamic weighting scheme to self-adapt its weights depending on the prediction scores allows the model to adjust for instances with varying levels of difficulty resulting in gradient updates driven by hard minority class samples. We further show that the proposed loss function leads to improved confidence calibration. Experiments conducted on highly imbalanced data across different applications of cyber intrusion detection (CICIDS2017 data set) and medical imaging (ISIC2019 data set) show robust generalization. Theoretical results supported by superior empirical performance provide justification for the validity of the proposed Dynamically Weighted Balanced (DWB) Loss Function.

### 1.2.2 Uncertainty Quantification in Deep Learning

While Deep Neural Networks (DNNs) are powerful algorithms capable of learning high level feature representations from data yielding high predictive accuracies, the model output in a standard neural network does not produce a measure for model uncertainty. Knowing the model confidence with which we can trust the output is desirable to explicitly process uncertain or ambiguous inputs. Quantification of uncertainty is imperative in safety critical applications ranging from medical diagnosis [3] to autonomous driving [4], [5] where the cost of error is high. For instance, an autonomous vehicle should not only identify the objects in its surrounding environment correctly, but it is equally important to provide a reliable measure of confidence on its predictions that allows approaching

areas of uncertainty with extra caution. In automated medical systems, the confidence of the network being deployed in the diagnosis of the disease should be properly estimated and notified to human specialists in the event of high uncertainty. Similarly, several other applications of Artificial Intelligence (AI) such as climate forecasting and investment decision making rely upon assessment of the uncertainty. Thus, the practical applicability of DL in real-world problems is enhanced if the models provide a quantifiable measure of certainty of its predictions. In chapter 4, we construct a deep probabilistic model to quantify uncertainty in deep learning and experimentally evaluate the proposed approach on complex neuro-imaging data.

### 1.2.3 Sparse Bayesian Time-to-Event Modeling and Radiomics

High-dimensional data is common in several scientific domains where a parsimonious description of the model with a lower-dimensional structure is often required. Bayesian framework is a potent alternative to the standard frequentist approach in this context due to its desirable sparsity inducing properties, the ability to incorporate domain knowledge, and easily accessible estimates of uncertainty. In chapter 5 we focus on survival prediction of patients with Glioblastoma involving high-dimensional quantitative radiomic features. We propose a sparse Bayesian approach that leverages shrinkage priors to address high-dimensionality problem in time-to-event modeling, more specifically, a Bayesian accelerated failure time (AFT) model with sparse properties. It is empirically validated on complex neuro-oncology MRI data (BRATS2020) to predict the survival of patients with Glioblastoma Multiform. The results suggest that image-based phenotyping provide incremental prognostic value over non-imaging clinical features. While the predictive ability of the sparse Bayesian approach proposed is in par with the standard approaches, it lends more insight into the prediction process through uncertainty information which is not readily available in a non-Bayesian setting.

## 1.3 Contributions

To summarise, this dissertation documents the following key contributions:

- We provide a differentiable loss formulation for class imbalance learning and demonstrate that the proposed approach allows to learn models that are already well calibrated. With experiments in interdisciplinary domains of cyber-security and medicine we show the proposed approach is superior to traditional methods.

- For safety critical systems uncertainty estimation is crucial. We provide methods for segmentation uncertainty quantification with probabilistic deep learning and show their applicability with experiments in clinical imaging.
- We present a sparse regression-based Bayesian accelerated failure time model where the sparsity is induced across predictors through a shrinkage prior. We identify relevant imaging-based prognostic factors that influence survival of patients with Glioblastoma. We demonstrate that inclusion of radiomic-based features enhance the predictive performance.

## 1.4 Dissertation Structure

The dissertation is structured as follows: Chapter 2 is concerned with providing deep learning preliminaries to establish the necessary background for the subsequent chapters. The following three chapters discuss our presented methods related to class imbalance, uncertainty and high-dimensionality: class imbalance learning and confidence calibration in Chapter 3, segmentation uncertainty in Chapter 4 and high dimensional time-to-event modeling in Chapter 5. In each chapter, we provide a review of literature, introduce formulations and empirically validate the methods with experiments in bio-medical imaging. Finally, Chapter 6 concludes the dissertation. In Chapter 6, we provide a discussion on limitations, critical challenges, and future directions.

## 1.5 Notation

Unless otherwise stated, the notation used throughout the dissertation is as follows. Matrices are represented by bold upper-case letters (e.g.,  $\mathbf{X}$ ), vectors are signified with bold lower-case letters (e.g.,  $\mathbf{x}$ ) and the scalars are represented by non-bold lower-case letters (e.g.,  $x$ ). Features (covariates) are denoted by  $\mathbf{x} \in \mathcal{X}$  where  $\mathbf{x}$  is a row vector and  $\mathcal{X}$  is the underlying population. Subscripts with bold letters denote entire rows/columns ( e.g.,  $\mathbf{x}_i$ ). We assume that the empirical data set  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  is formed by  $n$  independently and identically distributed (abbreviated by i.i.d.) draws from the population  $\mathcal{X}$ . The likelihood function is expressed by  $p(\mathbf{x}|\theta)$  where  $\theta \in \Theta$  are the model parameters from parameter space  $\Theta$ . We represent the data set likelihood by  $P(\mathbf{X}|\theta) = \prod_{i=1}^n p(\mathbf{x}_i|\theta)$  where the factorization is due to the i.i.d. assumption. In supervised learning tasks we denote the feature matrix by  $\mathbf{X}$  and the response vector by  $\mathbf{y} = \{y_1, \dots, y_n\}$ , and the likelihood function is a conditional model  $p(y|x, \theta)$  in which data set likelihood is represented by  $\prod_{i=1}^n p(y_i|\mathbf{x}_i, \theta)$ . Conditional distributions are expressed as  $P(\mathbf{X}|\theta)$  where  $\theta$  is a random variable,

but in the case where the parameters are fixed,  $P(\mathbf{X}; \theta)$  denotes the distribution. We use the notation  $\mathbb{H}$ ,  $\mathbb{E}$  and  $\mathbb{V}$  to represent the entropy, expectation and variance operators, respectively.



## CHAPTER 2

### DEEP LEARNING: PRELIMINARIES

In this chapter, we briefly provide statistical and mathematical underpinnings in deep learning. Starting with preliminaries in standard neural networks, we then discuss their Bayesian formulation to form the basis for the dissertation’s subsequent discussion on uncertainty estimation via probabilistic deep learning.

#### 2.1 Mathematical and Statistical Foundations in Deep Learning

The training paradigm of deep neural networks can be explained starting with linear models. Let a data set of size  $n$  be denoted by  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  where  $\mathbf{x}_i$  is the feature vector and  $y_i$  is the corresponding response. Let us first consider Generalized Linear Models (GLMs) [6] where the response is assumed to follow an exponential distribution. The structural form of the model is such that the expected value of the dependent variables is modeled through a transformation of inner product between the feature vector and parameters:

$$\mathbb{E}[y_i|\mathbf{x}_i] = g^{-1}(\mathbf{x}_i\mathbf{w} + b), \tag{2.1}$$

where  $\mathbb{E}[y|\mathbf{x}]$  is the conditional expectation,  $g: \mu \rightarrow \mathbb{R}$  is the link function that linearly connects the mean response to the covariates,  $\mathbf{w} \in \mathbb{R}^d$  is a parameter vector and  $b \in \mathbb{R}$  is a scalar.

However, features in its original form may not be sufficient to predict the corresponding response variable and one possible alternative is a feature transformation  $\tilde{\mathbf{x}} = h(\mathbf{x})$  where  $h(\cdot)$  is the transformation function. For instance,  $h(\cdot)$  can be specified by a polynomial expansion of the features, for e.g., in two-dimensional case  $h(\mathbf{x}) = (x_1, x_2, x_1^2, x_1x_2, x_2^2)$ . Feature transformation can be parameterized  $\tilde{\mathbf{x}} = h(\mathbf{x}; \theta)$ , where  $\theta$  are the parameters such that feature representations are extracted by the model which then leads to the adaptive basis function regression,  $\mathbb{E}[y_i|x_i] = g^{-1}(h(\mathbf{x}_i; \theta)\mathbf{w} + b)$ . Neural Networks can be considered as adaptive basis function regressors where a sequence of stacked GLMs signify the basis function:

$$\mathbb{E}[y_i|\mathbf{x}_i] = g^{-1}(h(\mathbf{x}_i; \mathbf{W}_1, \mathbf{b}_1)\mathbf{w}_2 + b_2), \quad (2.2)$$

where  $h(\mathbf{x}_i; \mathbf{W}_1, \mathbf{b}_1) = f(\mathbf{x}_i\mathbf{W}_1 + \mathbf{b}_1)$  is the feature transformation.

The success of deep neural networks is due to their ability to extract representative features and hidden structural knowledge from data automatically. We next briefly discuss feed-forward neural networks in the context of supervised classification.

### 2.1.1 Feed-forward Neural Networks

In the supervised setting, in a training set with  $n$  training examples  $\{\mathbf{x}_i, y_i\}_{i=1}^n$ , each input vector  $\mathbf{x}_i \in \mathbb{R}^n$  is associated with a corresponding class label (classification target)  $y_i \in \{1, \dots, c\}$ . Given a feature vector  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$  with  $d$  individual features  $x_i$ , a deep neural network with  $L$  hidden layers can be represented by a non-linear function  $f_\theta: \mathcal{X} \rightarrow \mathcal{Y}$  with model parameters  $\theta = \{\theta_1, \dots, \theta_L\}$ . Here,  $\theta_i = \{\mathbf{W}_i, \mathbf{b}_i\}$ , where  $\mathbf{W}_i$  is the weight matrix and  $\mathbf{b}_i$  is the bias vector for layer  $i$ . Then the DNN presents a complex feature transformation through  $a(\mathbf{x}) = f(\mathbf{W}_L \cdot f(\dots f(\mathbf{W}_2 \cdot f(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots) + \mathbf{b}_L)$ . Typically, the mapping function  $f(\cdot)$  consists of an affine transformation (either matrix multiplication or convolution) and a non-linear transformation (activation function). The general activation formula for the  $l^{\text{th}}$  layer in the  $j^{\text{th}}$  node can then be represented by:

$$a_j^{[l]} = f^{[l]}(\sum_k w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]}), \quad (2.3)$$

where  $a_j^{[l]}$  is the activation of the  $j^{\text{th}}$  neuron in the  $l^{\text{th}}$  layer,  $g^{[l]}$  is the activation function in the  $l^{\text{th}}$  layer,  $w_{jk}^{[l]}$  is the weight connection in the  $l^{\text{th}}$  layer from neuron  $j$  in  $(l-1)^{\text{th}}$  layer to neuron  $k$  in  $l^{\text{th}}$  layer and  $b_j^{[l]}$  is the bias term of the  $j^{\text{th}}$  node in  $l^{\text{th}}$  layer.

The feature vector in the last hidden layer is mapped to the output space  $\mathcal{Y}$  to obtain the network output which is passed through a softmax function to convert into normalized (pseudo) probabilities for different possible output classes. In a softmax layer with  $c$  neurons, the probability of class  $j$  given the feature vector  $\mathbf{x}$  is computed as:

$$P(y = j|\mathbf{x}) = \frac{\exp(a(\mathbf{x})^T \mathbf{W}_j^{[s]} + \mathbf{b}_j^{[s]})}{\sum_{j=1}^c \exp(a(\mathbf{x})^T \mathbf{W}_j^{[s]} + \mathbf{b}_j^{[s]})}, \quad (2.4)$$

where  $a(\mathbf{x})$  is the output of penultimate layer, and  $\mathbf{W}_j^{[s]}$  and  $\mathbf{b}_j^{[s]}$  are weights and bias terms in the  $j^{th}$  node connecting penultimate layer to the softmax layer,  $s$ .

To find the optimal model parameters, the network is then updated iteratively with respect to a loss function  $\mathcal{L}(f(\mathbf{x}_i; \theta), y_i)$  using an optimizer (traditionally, back-propagation algorithm):

$$\arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(\mathbf{x}_i; \theta), y_i), \quad (2.5)$$

where  $\theta$  represents model parameters,  $n$  is the sample size and  $\mathcal{L}$  is the loss function.

The predicted class label  $\hat{y}$  for any input instance  $\tilde{\mathbf{x}}$ , is the index of the maximum predicted score among all classes,  $\arg \max_j [P(y = j | \tilde{\mathbf{x}})]$ . The loss function for multi-class classification is usually the categorical Cross Entropy (CE) which is defined as:

$$\mathcal{L}_{CE}(\hat{y}, y) = - \sum_{j=1}^c y_j \log(\hat{y}_j), \quad (2.6)$$

where  $y_j = 1$  if training instance  $\mathbf{x}_i$  belongs to class  $c_j$  and 0, otherwise. Particularly, the objective function CE tries to maximize the likelihood of the target class for each training instance.

### 2.1.2 Convolutional Neural Networks

Deep learning models with different model architectures lend themselves to solve a large variety of problems. While 2D-CNN models have become the de facto standard for image processing applications, 1D-CNN models have shown to be effective in various applications in sequence processing such as anomaly detection [7], speech processing [8] and biomedical data classification [9].

The key attribute of neural networks is their ability to derive complex feature representations as linear combinations of the inputs which are then used to model the target as a non-linear function of the derived features. As in traditional machine learning, deep neural network based solutions do not require application of feature engineering techniques since the feature learning process is completed automatically. Through convolutional learning and spatial pooling operations, CNNs aggregate local features to extract complex hierarchical feature representations from feature sequence.

CNNs are composed of two distinct alternating layer types: convolutional and sub-sampling layers. The first convolutional layer in a CNN extract primitive features of network traffic while the subsequent convolutional layers can deduce more sophisticated features. The activation unit in a CNN represents the results of the convolution operation of the input data with a kernel.

The convolution layer is followed by a max-pooling layer for dimensionality reduction of data. Finally, the dense layer classifies the output classes combining all complex features identified by convolutional layers. An example of a CNN architecture is depicted in Figure 2.1.

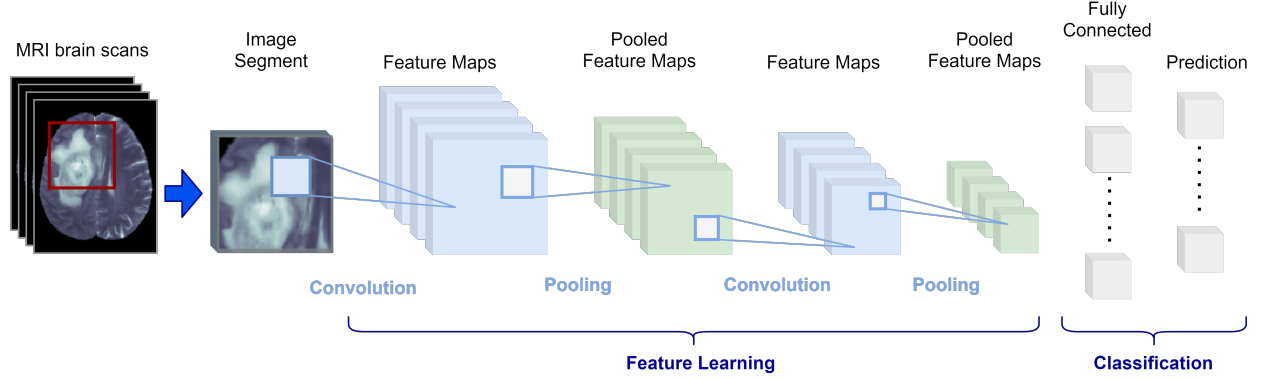


Figure 2.1.: Convolution and Max pooling layer operations in a CNN architecture. Figure depicts an example of patch-wise segmentation with CNN architecture. The input to the CNN is an image patch and the output is the probabilities for each class where the prediction for center pixel is the class with the highest score [1].

Feature map extraction using a one-dimensional convolution operation can be expressed as:

$$a_j^{l+1}(\tau) = \sigma \left( \sum_{f=1}^{F^l} K_{jf}^l(\tau) * a_f^l(\tau) + b_j^l \right), \quad (2.7)$$

where the feature map  $j$  in layer  $l$  is denoted by  $a_j^l(\tau)$ , non-linear function by  $(\sigma)$ , the number of feature maps in layer  $l$  by  $F^l$ , convolution kernels by  $K_{jf}^l$  and bias vector by  $b_j$ .

## 2.2 Probabilistic Deep Learning

To highlight the preliminaries in uncertainty estimation, here we provide the main concepts underlying the probabilistic representation of DL.

### 2.2.1 The Bayesian Interpretation of Neural Network Learning

Let  $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$  denote the training set of size  $n$  where each input vector  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$  is associated with a real-valued output, i.e.  $y \in \mathbb{R}$  in a regression task, or a class label  $y_i \in \{1, \dots, c\}$  in a classification setting. Deep neural nets are non-linear parametric function approximators. Given training inputs  $X = (x_1, \dots, x_n)$  and corresponding labels  $Y = (y_1, \dots, y_n)$ , the objective is to find the model parameters  $\omega$  of a function  $f_\omega: X \rightarrow Y$  which approximates the true data distribution. In a nutshell, a neural network models the likelihood  $p(y|\mathbf{x}; \omega)$  as a nonlinear function

of  $\omega$  and  $\mathbf{x}$ . A standard neural network optimizes the weights and biases  $\omega = \{W_l, b_l\}_{l=1}^L$  (network parameters) and finds a single weight assignment or point estimate  $\omega^*$  by minimizing the expected loss:

$$\min_{\omega} \mathbb{E}_{\mathbf{x}, y}[\mathcal{L}(f(\mathbf{x}; \omega), y)] \approx \min_{\omega} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(\mathbf{x}_i; \omega), y). \quad (2.8)$$

Alternatively, from a probabilistic perspective, this is equivalent to maximum Likelihood Estimation (MLE) which seeks to maximize the likelihood of data given the weights:

$$\max_{\omega} \mathbb{E}_{\mathbf{x}, y}[\log p(D|\omega)] \approx \max_{\omega} \sum_{i=1}^n \log p(y_i|x_i; \omega). \quad (2.9)$$

While neural networks typically perform well, they tend to produce over-confident predictions and does not produce a coherent measure for uncertainty urging a shift towards probabilistic deep learning. In the Bayesian probabilistic interpretation of deep neural networks, each model is specified by a prior and likelihood. Model averaging, also known as marginalization, is the key feature in the Bayesian approach. Instead of directly optimising model weights and inferring point estimates, the Bayesian extension of a Neural Network learns a probability distribution over the weights by averaging over all possible model weights (Figure 2.2).

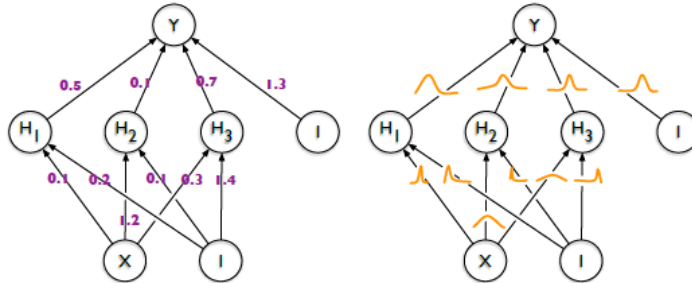


Figure 2.2.: Left: Deterministic neural network where Weights are point estimates. Right: Probabilistic neural network where each weight is sampled from a probability distribution. [10]

(a) *Bayesian Learning of the weights:*

In the Bayesian framework, a prior distribution  $p(\omega)$  is placed over the space of parameters  $\omega \in \Omega$ , which is then updated according to the Bayes rule based upon the observed data to obtain the posterior distribution. Prior distribution captures our prior belief on what the parameters of the

model are, whereas the likelihood represents  $p(Y|X, \omega)$  the probability that the data is observed given some parameter  $\omega$ . The posterior  $p(\omega|X, Y)$  is the distribution over possible values of  $\omega$  which allows to obtain the most probable function parameters based on observations. Posterior distribution is computed as:

$$p(\omega|X, Y) = \frac{p(Y|X, \omega) p(\omega)}{p(Y|X)}, \quad (2.10)$$

where  $p(Y|X)$  is the model evidence which marginalize the likelihood over  $\omega \in \Omega$ .

The marginal likelihood integral  $p(Y|X) = \int p(Y|X, \omega) p(\omega) d\omega$  is intractable, and therefore the true posterior density  $p(\omega|X, Y) = p(Y|X, \omega) p(\omega)/p(Y|X)$  is intractable.

Typically, a standard matrix Gaussian prior is assumed over the weights, or  $W \sim \mathcal{N}(0, \mathbb{I})$ . Let the random output of the Bayesian neural network (BNN) be defined as  $f^\omega(\mathbf{x})$ . For a regression task, the likelihood is typically a Gaussian  $p(y|f^\omega(\mathbf{x})) = \mathcal{N}(y; f^\omega(\mathbf{x}), \sigma^2)$  with distribution parameters mean ( $\mu$ ) and variance ( $\sigma^2$ ) equal to the model output  $f^\omega(\mathbf{x})$  and observation noise, respectively. In classification, model output is obtained via softmax function,  $p(y|f^\omega(\mathbf{x})) = \text{softmax}(f^\omega(\mathbf{x}))$ . Thus, the prior distribution can be determined based on the a priori known information and the likelihood can be computed given the observed data. However, the calculation of posterior is computationally expensive.

*(b) Posterior predictive distribution*

Each possible parameter configuration weighted based on the posterior distribution allows to obtain the prediction of an unknown label. More formally, the predictive distribution of an output  $y$  of a test input  $x^*$  is obtained by computing a weighted expectation over all possible model parameters  $\Omega$  as follows:

$$p(y^*|x^*, X, Y) = \mathbb{E}_{p(\omega|X, Y)}[p(y^*|x^*, \omega)] \quad (2.11)$$

$$p(y^*|x^*, X, Y) = \int_{\Omega} \underbrace{p(y^*|x^*, \omega)}_{\text{Data}} \underbrace{p(\omega|X, Y)}_{\text{Model}} d\omega, \quad (2.12)$$

where  $\mathbb{E}_{p(\omega|X, Y)}[p(y^*|x^*, \omega)]$  denotes the expectation of  $[p(y^*|x^*, \omega)]$  with respect to  $p(\omega|X, Y)$  and  $p(y^*|x^*, \omega)$  is the conditional predictive distribution and  $p(\omega|X, Y)$  is the posterior over parameters.

In equation 2.12, data uncertainty is represented by the posterior distribution over model output  $y$  given model parameters  $\omega$ . The posterior distribution over parameters given data characterizes the model uncertainty [11].

While Bayesian inference amounts to conditioning on data and determining the posterior distribution  $p(\omega|X, Y)$  of the model parameters, the computation of model evidence in the denominator in posterior involves an integration over the high dimensional parameter space, which is often analytically intractable. The true posterior  $p(\omega|X, Y)$  is therefore approximated via a variational distribution  $q_\theta(\omega)$ , parameterised by  $\theta$  [10, 12, 13].

Since the integral in Eq. 2.12 is also intractable, it is typically approximated through sampling strategies [14, 15] as follows:

$$p(y^*|x^*, X, Y) \approx \frac{1}{M} \sum_{m=1}^M p(y^*|x^*, \omega_m), \quad \omega_m \sim q(\omega|X, Y), \quad (2.13)$$

where  $\{p(y^*|x^*, \omega_m)\}_{m=1}^M$  is an ensemble of  $M$  models and  $\omega_m$  are samples from an approximate posterior distribution  $q(\omega|X, Y)$ .

### 2.2.2 Variational Inference

*Inference as optimization:* Variational Inference (VI) [16] involves approximating the intractable true posterior distribution  $p(\omega|X, Y)$  with an approximating variational distribution  $q_\theta(\omega)$ , parameterised by  $\theta$ . The similarity between two distributions can be measured by the Kullback-Leibler (KL) divergence [17], which is computed as follows:

$$KL(q_\theta(\omega)||p(\omega|X, Y)) = \int_{\Omega} q_\theta(\omega) \log \frac{q_\theta(\omega)}{p(\omega|X, Y)} d\omega. \quad (2.14)$$

The optimal approximation is the setting of the parameters  $\theta$  that minimize the KL divergence between the  $q_\theta(\omega)$  and posterior distribution of interest:

$$\theta^* = \arg \min_{\theta} KL[q_\theta(\omega)||p(\omega|X, Y)]. \quad (2.15)$$

Applying the Bayes rules allows us to re-write this optimization as:

$$\begin{aligned}
\theta^* &= \arg \min_{\theta} \int_{\Omega} q_{\theta}(\omega) \log \frac{q_{\theta}(\omega) p(Y|X)}{p(Y|X, \omega) p(\omega)} d\omega \\
&= \arg \min_{\theta} \log q_{\theta}(\omega) - \log p(\omega) - \log p(Y|X, \omega) \\
&= \arg \min_{\theta} KL[q_{\theta}(\omega)||p(\omega)] - E_{q_{\theta}(\omega)} \log P(Y|X, \omega),
\end{aligned} \tag{2.16}$$

where  $q(\omega|\theta)$  is the variational posterior,  $p(\omega)$  is the prior and  $p(D|\omega)$  is the likelihood. Since  $\log p(Y|X)$  is a constant, it is omitted from the optimization.

*Variational objective:* The variational objective of minimizing KL divergence is equivalent to maximizing the variational lower bound, also called evidence lower bound (ELBO):

$$\theta^* = \arg \max_{\theta} E_{q_{\theta}(\omega)} \log P(Y|X, \omega) - KL[q_{\theta}(\omega)||p(\omega)]. \tag{2.17}$$

Maximizing the first term in ELBO, referred to as expected log likelihood with regard to  $q_{\theta}(\omega)$  enables obtaining  $q_{\theta}(\omega)$  which explains the data best. The second term acts as a regularization term, and aims to minimize KL divergence between the variational distribution and prior.

Then at inference, an approximate predictive distribution can be derived using the variational predictive distribution:

$$p(y^*|x^*, X, Y) = \int_{\Omega} p(y^*|x^*, \omega) q_{\theta^*}(\omega) d\omega, \tag{2.18}$$

where  $\omega = \{W\}_{i=1}^L$  is the set of weights for a neural network with  $L$  layers.

### 2.2.3 Bayesian Convolutional Neural Networks

We conclude this chapter by extending the previously discussed approximate Bayesian inference approach to CNNs which will facilitate obtaining model uncertainties for imaging data. Consider the CNN architecture discussed in section 2.1.2 and in Figure 2.1. More formally, assume a three-dimensional tensor  $x \in \mathbb{R}^{H_{i-1} \times W_{i-1} \times K_{i-1}}$  is fed into the  $i^{th}$  convolutional layer, where  $H_{i-1}$  is the height,  $W_{i-1}$  is the width and  $K_{i-1}$  are the channels. Each convolution layer is consisted of a series of filters known as kernels, denoted by  $\mathbf{k}_k \in \mathbb{R}^{h \times w \times K_{i-1}}$  for  $k = 1, \dots, K_i$ , where  $h$ ,  $w$  and  $K_{i-1}$  denote the kernel width, height, and number of channels, respectively. The convolution of kernels with the input at a specified stride  $s$  yields an output layer



with dimension equal to  $y \in R^{H'_{i-1} \times W'_{i-1} \times K_i}$ , where  $H'_{i-1}, W'_{i-1}$  and  $K_i$  are the new height, new width and the number of kernels, respectively. The sum of element-wise multiplication between kernel  $\mathbf{k}_k$  with a corresponding input patch yields the elements  $y_{i,j,k}$  in the output layer:  $[[x_{(i-h/2, j-w/2, 1)}, \dots, x_{(i+h/2, j+w/2, 1)}], \dots, [x_{(i-h/2, j-w/2, K_{i-1})}, \dots, x_{(i+h/2, j+w/2, K_{i-1})}]]$ .

A Probabilistic CNN allows to represent the parameter uncertainty in the form of probability distributions (Figure 2.3). A standard CNN can be transformed into its probabilistic formulation by placing a prior over each kernel. Posterior is analytically intractable, and each kernel-patch pair can be integrated via approximate variational inference methods discussed above. More on probabilistic deep learning and its application to uncertainty estimation will be discussed in Chapter 4.

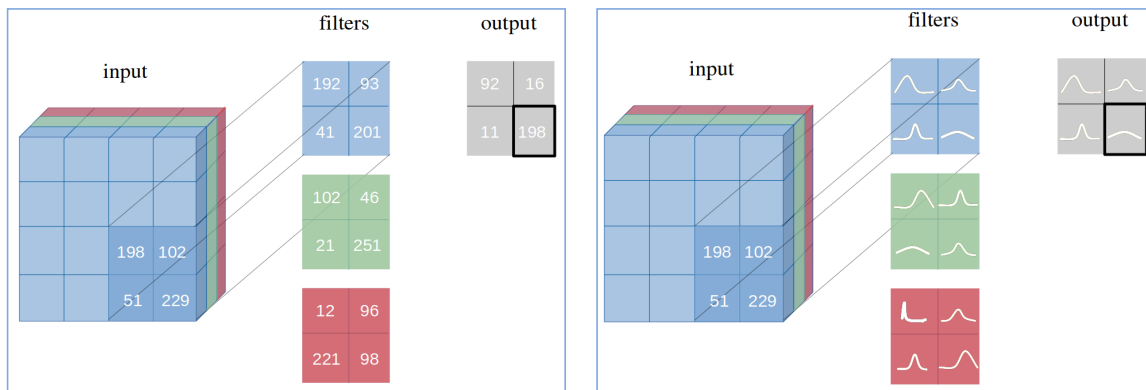


Figure 2.3.: An example of a BCNN. Input image, filters and output are depicted in the figure. Left: each weight has a single fixed value. Right: each weight is represented by a probability distribution. [18]

**CHAPTER 3**  
**DYNAMICALLY WEIGHTED BALANCED LOSS: CLASS IMBALANCED**  
**LEARNING AND CONFIDENCE CALIBRATION OF DEEP NEURAL**  
**NETWORKS** <sup>1</sup>

### **3.1 Introduction**

With Artificial Intelligence, mobile and Internet of Things (IoT) driving data complexity and new sources of data, a new paradigm named big data has emerged. High class imbalance, often observed in large-scale data sets [19], occurs when some classes are under-represented (minority) compared with few classes that dominate (majority), introducing a distributional bias in favor of the majority classes. Such a skewed class distribution with a biased learning process could result in underestimation of minority class conditional probabilities hindering classification performance. Despite decades of research, training unbiased models from highly imbalanced data sets continues to be an open problem.

Data distribution imbalance is predominant in many real-world classification tasks, such as fault diagnosis [20], [21], network intrusion detection [22], medical diagnosis [23], [24], electricity pilferage [25] and fraudulent transactions [26], among others. Handling class-imbalance is of great importance in these situations, where the minority class is the class of interest with respect to the learning task. For instance, a malicious program should not be misclassified as benign which could lead to more adverse consequences than the reverse. Similarly, a malignant skin lesion which is rare should still be correctly identified. The same applies to several other application areas, where the accurate detection of rare events is crucial. As conventional classifiers rely on balanced class distributions, they will tend to misclassify minority observations of data with a skewed class distribution [27], [28]. Thus, a classifier which perform well and learn effectively from inherently more difficult and rare classes is highly desirable.

---

<sup>1</sup>Portions of this chapter have been previously published in IEEE Transactions on Neural Networks and Learning Systems (2021): <https://doi.org/10.1109/TNNLS.2020.3047335>, and have been reproduced with permission from IEEE Publishing.

Obtaining reliable probability estimates is crucial to make informed decisions in real-world applications. This is even more necessary for imbalanced data due to high uncertainty around rare events. While classification of data featuring high class imbalance has received attention in prior research, reliability of class membership probabilities in the presence of class imbalance has been previously assessed only to a very limited extent [29], [30]. A closer look to the previous studies on probability calibration shows that research on classification calibration under class imbalance in the context of deep learning is so far lacking in the scientific literature.

Deep Learning (DL) has arguably become the most crucial breakthrough in machine learning and has achieved the state-of-the-art performance in various applications. Deep Neural Networks (DNNs) are comprised of sums of non-linearly transformed linear models [31] and, thus, are trained to approximate non-linear functions between the input and output. In neural networks, the feedback generated by the loss function helps in optimizing the parameters. Due to the feasibility in differentiable optimization, the most common choice for the loss function in multi-class classification is the cross entropy. Classical cross-entropy based loss function gives equal importance for each data instance, which will lead the network oversee classes with fewer number of observations. Thus, cross entropy loss is improper in classification or segmentation tasks under class imbalance. A simple heuristic method which is widely adopted to balance loss in the presence of class imbalance is to set class weights inversely proportional to the class frequency [21], [32]. However, this strategy reveals poor performance on large-scale real-world data. In contrast, we propose a dynamic strategy to assign class weights with emphasis on hard to train instances and propose a novel loss function called *Dynamically Weighted Balanced (DWB) Loss* which is capable of naturally handling the class imbalance while also leading to improved calibration performance. To illustrate the generality of the proposed approach, experiments are conducted on challenging real-world applications in cyber intrusion detection and skin lesion diagnosis.

*Contributions:* The proposed approach is distinct in two ways with respect to the previous work: (1) Instead of a fixed weighting scheme, the assigned weights self-adapts its scale based on the prediction difficulty of the data instance. (2) We link class imbalance and reliability of confidence estimates. To the best of our knowledge, prior research has not addressed both these issues in a unified approach in the context of deep learning. The paper therefore presents the following major novelties: (1) A differentiable loss formulation based on a class rebalancing strategy, where the weights are dynamically changed during the course of training. (2) A framework that allows to

learn models that are already well calibrated, thus simultaneously addressing both class imbalance and reliability of class membership probabilities in deep neural networks.

## 3.2 Related Work

### 3.2.1 Class Imbalance

Despite recent advances in deep learning, the research on deep neural networks to address class imbalance remain limited [33]. We briefly describe below the traditional methods and prominent work in recent years on deep imbalanced learning.

Most of the previous efforts to handle class imbalance can be divided into two categories: data-level and algorithmic-level methods. *Data-level* methods [34–41] alter the class distribution in the original data by employing re-sampling strategies to balance the data set. The simplest forms of re-sampling include random over-sampling and random under-sampling. The former handles class imbalance by duplicating the instances in the rare minority class and thus, augmenting the minority class, whereas the latter randomly drops instances from the majority class to match the cardinality of minority class. Experiments conducted in [42] suggest that data sampling strategies have little effect on classification performance, however, results in [43] demonstrate that random over-sampling leads to performance improvements. While sampling strategies are widely adopted, these methods manipulate the original class representation of the given domain and introduce drawbacks. Particularly, over-sampling can potentially lead to overfitting and may aggravate the computational burden while under-sampling may eliminate useful information that could be vital for the induction process. Moreover, a classifier developed by employing sampling methods to artificially balance data may not be applicable to a population with a much difference prevalence rate since the classifier is trained to perform well on balanced data.

*Algorithm-level* approach involve adjusting the classifier, and can further be categorized into *ensemble methods* and *cost-sensitive methods*. The most widely used methods include bagging [44] and boosting [45] ensemble-based methods. Boosting algorithms such as AdaBoost work by placing more emphasize on harder to train examples and using them to train subsequent classifiers. Experiments in [46] suggests boosting performs better than sampling methods. Alternatively, hybrid ensemble methods which combine sampling and boosting methods [47], [48] have also been proposed in past literature. A thorough review on ensemble techniques for imbalanced data with emphasis

on two-class problems is presented in [49]. While ensemble-based algorithms are worthwhile, the use of multiple classifiers makes them more complex which leads to increased training times.

To reinforce the sensitivity of the classification algorithm towards the under-represented class, *cost sensitive learning* methods incorporate class-wise costs into the objective function of the classification algorithm during training process. Cost parameters can be arranged in the form of a cost matrix such that higher costs are associated with misclassification of an observation from the minority class [50]. However, design of the cost matrix which includes different misclassification costs associated with each class sample may require expert judgement. Another approach to cost-sensitive learning is rescaling the data, performed by assigning training examples of different classes with different weights (re-weighting), re-sampling the training instances or shifting the decision threshold based on their misclassification costs. These methods have been reported to perform well on binary data [50]. In [28], authors study techniques which are proven to be efficient in handling class imbalance. They conclude that while almost all methods are effective on binary classification, some methods are only effective in binary case and that cost sensitive learning can become highly complicated in multi-class setting.

Among recent contributions in *deep imbalanced learning*, Khan et al. [51] proposed a cost sensitive approach where they optimized both the model parameters and cost parameters synchronously. In the domain of computer vision, a recently proposed loss function called Focal Loss [52] for object detection attracted considerable attention in which they promote harder samples by down-weighting the loss assigned to well-classified instances. A meta-learning approach that determines per-sample loss weights of the training data based on their gradient directions is presented in [53], but requires an additional validation set and takes approximately three times the training time compared to regular training. Zhang et al. [54] proposed an evolutionary cost-sensitive deep belief network (ECS-DBN) to improve the imbalance classification performance of Deep Belief Networks (DBN). However, their approach is prohibitively expensive since the class-dependent misclassification costs are first optimized by an adaptive differential evolution algorithm (EA). A method that combines hard sample mining with a newly introduced class rectification loss (CRL) function is proposed in [55]. They adopt a batch-wise hard sample mining approach on the minority class. In [56], loss reweighting is performed by the inverse effective number of samples. Based on the assumption that the samples with too many similar gradient norms are the easy samples, authors in [57] suggested a counting based approach called Gradient Harmonizing Mechanism (GHM).

Current approaches for handling class imbalance in deep learning contains drawbacks with respect to over-fitting, loss of information, complexity and require changes to the network architectures and optimization process. Furthermore, existing methods for loss reweighting require careful tuning of hyper-parameters which can be computationally expensive.

### 3.2.2 Confidence Calibration

Most of the previous studies have almost exclusively focused on either class imbalance or obtaining calibrated probability estimates, but handling both these issues concurrently remains briefly addressed in literature [29], [30]. In [29], authors show that probability estimates of the instances in minority classes are unreliable and that the methods of handling class imbalance do not automatically address calibration. Moreover, experiments in [30] demonstrates that strategies adopted to mitigate effects of class imbalance such as under-sampling adversely affect probability calibration of minority classes. In the context of neural networks, post-hoc calibration methods including matrix scaling, vector scaling and temperature scaling are widely adopted for probability calibration. Temperature scaling, proposed more recently by Guo et al. [58] gained significant attention. The method is applied to the logits of the neural network and require a validation set to tune a temperature parameter. However, the performance of these approaches in the presence of class imbalance is not adequately explored.

Differing from previous methods which require a handcrafted cost matrix, assign fixed weights, or involve algorithmic modifications, we propose a loss function incorporating a dynamic weighting factor adjusted during the training process to address training bias of imbalanced data which also result in well-calibrated confidence estimations. It does not require any additional hyper-parameter tuning and can be promptly applied to any deep neural network architecture.

## 3.3 Dynamically Weighted Balanced (DWB) Loss

### 3.3.1 Loss Function Formulation

#### Revisiting Categorical Cross Entropy

Let the training set with  $n$  samples be denoted by  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ , where  $\mathcal{X} \subset \mathbb{R}^{d_x}$  is the feature space and  $\mathcal{Y} \subset \mathbb{R}^{d_y}$  is the label space. For each data instance  $i$ ,  $\mathbf{x}_i \in \mathcal{X}$  is the input feature vector and  $y_i \in \mathcal{Y} = \{1, 2, \dots, c\}$  is the ground-truth class label. Consider a hypothesis (classifier) from a parametric family  $\mathcal{F} := \{f_\theta: \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} | \theta \in \Theta\}$  which maps input feature space

to the label space  $f : \mathcal{X} \rightarrow \mathcal{Y}$  and learns by minimizing the loss  $\mathcal{L}(f(\mathbf{x}; \theta), y)$ . Given a loss function  $\mathcal{L} : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}^+$  and a classifier  $f$ , the (empirical) risk is defined as  $R_L(f) = E_D[f(\mathbf{x}; \theta), y]$ , where the expectation is with respect to the the empirical distribution,  $D$ .

Consider a DNN with the softmax output layer with loss as the categorical cross entropy. Then the parameters of DNN can be optimized with empirical risk minimization where risk is defined as:

$$R_L(f) = E_D[f(\mathbf{x}; \theta), y_{\mathbf{x}}] = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^c y_{ij} \log(f_j(\mathbf{x}_i; \theta)), \quad (3.1)$$

where  $\theta$  is the set of parameters of the classifier,  $y_{ij}$  is the  $j^{\text{th}}$  element of one-hot encoded label of the instance  $\mathbf{x}_i$  with  $\mathbf{y}_i = e_{y_i} \in \{0, 1\}^c$  such that  $\mathbf{1}^T \mathbf{y}_i = 1, \forall i$ , and  $f_j(\mathbf{x}; \theta) \in \mathbb{R}^c$  is the model output with  $f_j$  denoting the  $j^{\text{th}}$  element of  $f$ . Since the output layer is a softmax,  $\sum_{j=1}^c f_j(\mathbf{x}_i; \theta) = 1$  and  $f_j(\mathbf{x}_i; \theta) \geq 0, \forall j, i, \theta$ .

### Dynamic Weighting of Loss Function

The backpropagation of error algorithm which is typically used to train neural networks updates the weights of the model in proportion to the errors made during training. As the misclassification errors of data instances from each class are given the same importance, for severely skewed class distributions this results in adapting the classifier in favor of majority class. While class imbalance does not hinder model performance in simple classification tasks with clear class separation, it affects classes that are inherently more difficult to classify. Training samples from classes with fewer observations producing lower class probabilities are expected to be the harder instances. Moreover, correct classifications tend to have greater softmax probabilities than those misclassified and out-of-distribution instances [59]. In this context, we introduce a dynamic weighting based classifier objective function based on the prediction probability of ground truth class to assign higher weights to hard to train instances, which we term the Dynamically Weighted Balanced (DWB) Loss. Let  $f_j(\mathbf{x}_i; \theta)$  be indicated by  $p_{ij}$  for convenience. Thus,  $p_{ij}$  is the predicted probability of the class  $j$  of instance  $\mathbf{x}_i$ . We define Dynamically Weighted Balanced (DWB) Loss as:

$$L_{DWB} = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^c w_j^{(1-p_{ij})} y_{ij} \log(p_{ij}) - p_{ij}(1 - p_{ij}), \quad (3.2)$$

where  $w_j$  is the class weight of class  $j$ ,  $y_{ij}$  is the  $j^{\text{th}}$  element of one-hot encoded label of instance  $\mathbf{x}_i$  and  $p_{ij}$  is the predicted probability of the class  $j$  of instance  $\mathbf{x}_i$ .

The proposed loss function is composed of two terms: dynamically weighted cross entropy and a regularization component equal to the entropy of brier score which can be considered as a reliability component that leads to better calibration (more on calibration is in section 3.3.2).

The class weights  $w_j$  can be handled as a hyper-parameter that is learned from data by cross validation or set proportional to inverse class frequency. We set  $w_j$  equal to the log ratio of the class frequency of the majority class and the class frequency  $n_j$  (computed over the training data set) as follows:

$$w_j = \log \left( \frac{\max(n_j | j \in c)}{n_j} \right) + 1. \quad (3.3)$$

As such, misclassification errors for a class  $j$  with class-wise cost of  $w_j$  will have  $w_j$ -times more penalty than misclassification errors for the majority class with weight equals to 1. For extremely imbalance classes, log smooths the weights and to avoid major class weight being less than 1, we add 1 to the log weights.

While a fixed-weighting approach based on class frequency balances the contribution from majority and minority classes, it does not discriminate between the easy and hard sample instances. Instead, we apply class-wise weights of various magnitudes from the same class depending on the prediction output and adjust the relative contribution of mispredictions. The loss function defined in equation (3.2) optimizes a dynamically weighted training loss which reflects labels' importance level based on class frequency while promoting hard positives which are predictions with low confidence scores.

For illustration purposes, we consider a case where class weight or class imbalance ratio,  $w_j = 2$ . Figure 3.1 provides an intuitive comparison of different losses: standard binary Cross Entropy (CE), cross entropy with fixed class weights set to imbalance ratio, Focal Loss (FL) and proposed Dynamically Weighted Balanced (DWB) Loss. It depicts how the proposed DWB loss reshapes the loss function based on the prediction probability of the target by dynamically assigning the importance weights. Note that the Focal Loss always produces a lower loss value when compared with the standard cross entropy loss. This results in FL still down-weighting correct predictions with low prediction scores ( $p < 0.6$ ). On the contrary, the proposed DWB loss penalizes more than the cross entropy if the predictions defined from the network outputs are confident and wrong.

We note two properties of the DWB Loss: (1) When a training instance is misclassified and  $p_{ij}$  is small, the loss is up-weighted. (2) As  $p_{ij}$  goes close to 1, the weighting factor for well classified



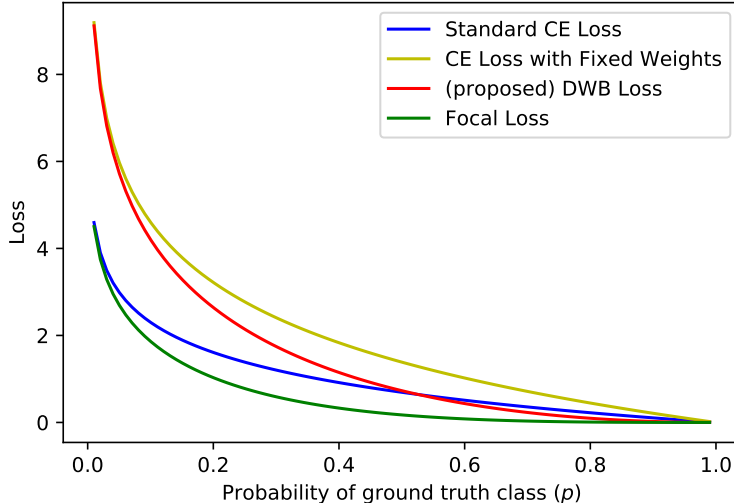


Figure 3.1.: Comparisons among proposed Dynamically Weighted Balanced (DWB) Loss and other commonly used losses for classification: the standard Cross Entropy (CE) loss, cross entropy with fixed weights assigned and the Focal Loss (FL) with hyper-parameter( $\gamma$ ) set to 2 (recommended). DWB Loss put more focus on hard to train, misclassified examples through a dynamic weighting factor.

instance is close to 1, hence the loss is unaffected and equivalent to Cross Entropy. Differing from FL which down-weights the contribution of easy samples, proposed DWB Loss focus more on hard examples by up-weighting the misclassified examples while taking into account both sample difficulty and the class frequency. Experiments suggest that the performance of the proposed loss function is superior to the previous class balancing approaches, implying that it is a more effective alternative to the existing methods.

We visualize dynamic class weights (dash lines) in Figure 3.2 for each class in imbalanced CI-CIDS2017 data set assigning different predicted probabilities for ground truth. Note that  $p_{ij} = 1$  corresponds to no re-weighting and  $p_{ij} = 0$  corresponds to re-weighting by imbalance ratio ( $w_j$ ) which is proportional to inverse class frequency (logarithm was not taken when computing the weighting factor in Figure 3.2 for better illustration). Thus, the introduced self-adapting weighting scheme enables smooth adjustment of the class-balanced term between re-weighting and no re-weighting of objective function. The proposed cost sensitive learning approach with DWB loss function is presented in Algorithm 1.

### 3.3.2 Improving Calibration using DWB loss

Biased training data with a skewed class distribution typically under-estimates the class probability estimates of minority class instances [29], and therefore, the predicted class probabilities are

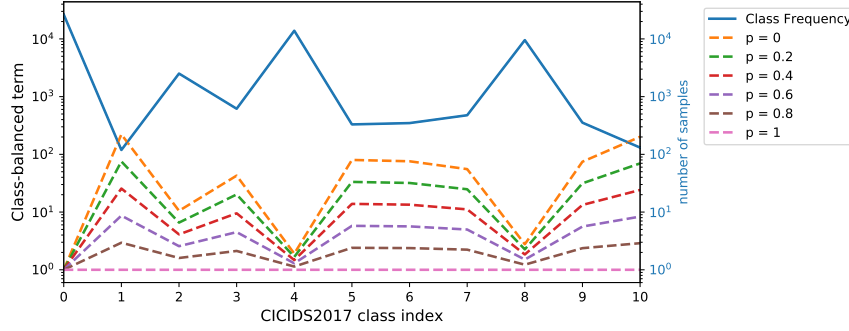


Figure 3.2.: Visualization of the weight term based on predicted probability of ground truth class ( $p$ ) on long-tailed CICIDS2017 data. Y-axis is in log scale. Solid blue line represents the number of samples in each class while the dash line represents how the assigned weight changes w.r.t prediction probability of ground truth class. Note that here we have not taken logarithm when computing the weighting factor for better visualization.

---

**Algorithm 1** Cost Sensitive Learning with DWB Loss

---

**Input:** Training set  $(\mathbf{X}_T, \mathbf{y}_T)$ , Maximum number of epochs  $M$ , Learning rate for  $\theta$ , Class weights  $w$ .

**Output:** Learned network parameters  $\theta^*$

- 1:  $\text{Net} \leftarrow \text{construct\_DNN}()$
  - 2:  $\theta \leftarrow \text{initialize\_Net}(\text{Net})$  ▷ Random Initialization
  - 3: **for**  $b \in [1, B]$  **do** ▷ Number of batches
  - 4:   **for**  $i \in [1, t]$  **do** ▷ Number of samples in batch
  - 5:      $p_{ij} \leftarrow \text{Forward}(\mathbf{x}_i, \mathbf{y}_i, \text{Net}, \theta)$  ▷ predicted probability of the class  $j$  of instance  $\mathbf{x}_i$
  - 6:      $w_{ij} \leftarrow (w_j)^{(1-p_{ij})}$
  - 7:   **end for**
  - 8:    $\mathcal{L} \leftarrow \text{compute\_loss}$  ▷ Eq. 3.2
  - 9:    $\nabla\theta \leftarrow \text{Backward}(\mathcal{L}, \mathbf{x}_b, \mathbf{y}_b, \text{Net}, \theta)$
  - 10:    $\theta^* \leftarrow \text{update\_NetParams}(\text{Net}, \theta, \nabla\theta)$
  - 11: **end for**
  - 12: **return**  $\theta^*$
- 

unreliable in class imbalance scenarios. The parameter estimation bias under class imbalance also applies to models which typically produce calibrated probability estimates, such as logistic regression [60]. Obtaining well-calibrated probability estimates which are reflective of the true likelihood of events [61] is highly desirable in real-world applications. The calibrated prediction probabilities are in concordance with the true occurrence of the event of interest and perfect calibration is formally defined as:

$$\mathbb{P}(Y = y | \hat{p} = p) = p \quad ; \forall p \in [0, 1], \quad (3.4)$$

where  $Y$  is a class prediction and  $\hat{p}$  is its associated confidence.

The regularizing component of the DWB loss is equal to the entropy of conditional distribution  $p = p_\theta(y|\mathbf{x})$  in Brier Score. Recall that entropy of a probability assignment is a measure of inherent uncertainty [62]. Below we show that the DWB Loss minimizes a regularized upper bound on the weighted Kullback-Leibler (KL) Divergence [17] between the true distribution  $\mathbf{q}$  and the predicted distribution  $\mathbf{p}$ .

Considering a data instance with class label  $y$ , ground truth probability  $q_y$  and class membership probability estimate  $p_y$ , we proceed to obtain the following:

$$\begin{aligned}
L_{DWB} &= -w_y^{(1-p_y)} q_y \log(p_y) - p_y (1 - p_y) \\
&\geq -w_y (1 - p_y) q_y \log(p_y) - p_y (1 - p_y) \\
&\quad ; \forall y, w_y \geq 1 \text{ and } p_y \in [0, 1] \\
&= -w_y q_y \log(p_y) - w_y |p_y q_y \log(p_y)| - p_y (1 - p_y) \\
&\quad ; \forall y, \log(p_y) \leq 0 \\
&\geq -w_y q_y \log(p_y) - \max(q_y) w_y |p_y \log(p_y)| \\
&\quad - p_y (1 - p_y) \\
&\geq -w_y q_y \log(p_y) + w_y p_y \log(p_y) - p_y (1 - p_y) \\
&\quad ; \forall y, q_y \in [0, 1] \\
&\geq \mathbf{w} (CE(\mathbf{q}, \mathbf{p}) - H(\mathbf{p})) - \mathbf{p} (\mathbf{1} - \mathbf{p}), \tag{3.5}
\end{aligned}$$

where  $CE(\mathbf{q}, \mathbf{p})$  is the cross entropy between true distribution  $\mathbf{q}$  and predicted distribution  $\mathbf{p}$ , and  $H(\mathbf{p})$  is the entropy of  $\mathbf{p}$ .

Since,  $CE(\mathbf{q}, \mathbf{p}) = KL(\mathbf{q}||\mathbf{p}) + H(\mathbf{q})$ , the above inequality can be represented as:

$$\begin{aligned}
L_{DWB} &\geq \mathbf{w} (KL(\mathbf{q}||\mathbf{p}) + \underbrace{H(\mathbf{q})}_{\text{constant}} - H(\mathbf{p})) - \mathbf{p} (\mathbf{1} - \mathbf{p}) \\
&\geq \mathbf{w} (KL(\mathbf{q}||\mathbf{p}) - H(\mathbf{p})) - \mathbf{p} (\mathbf{1} - \mathbf{p}), \tag{3.6}
\end{aligned}$$

where  $KL(\mathbf{q}||\mathbf{p})$  represents the KL divergence between target  $\mathbf{q}$  and predicted  $\mathbf{p}$  distributions.

The proposed loss constructs an upper bound on the weighted KL divergence with an additional regularization equal to the sum of  $\mathbf{w} H(\mathbf{p})$  and  $\mathbf{p}(\mathbf{1} - \mathbf{p})$ . While it seeks to minimize the deviation of the predicted distribution from the true label distribution through KL divergence, it aims to maximize the entropy terms, thereby penalizing over-confident predictions on the target as a form of

regularization which leads to better calibration. While FL has shown to have calibration properties in [63], we did not observe significantly improved results with it in our experiments.

### 3.3.3 DWB Loss Function Gradients

Let the predicted (unnormalized) output from the model be denoted by  $z_i$ , where  $i \in \{1, \dots, c\}$ . The softmax function  $\mathbb{R}^c \rightarrow \mathbb{R}^c$ , maps a vector  $z \in \mathbb{R}^c$  to a vector  $p \in \mathbb{R}^c$  which can be expressed as:

$$p_i(z) = \frac{e^{z_i}}{\sum_{j \in \{1, \dots, c\}} e^{z_j}} \quad ; \forall i \in \{1, \dots, c\} \quad (3.7)$$

where  $z$  is a real vector.

Given that for a data instance with class label  $y$ , the only non-zero element of the one-hot encoded vector  $\mathbf{y}$  is at the  $y$  index, the DWB loss is simplified as:

$$L_{DWB} = -w_y^{(1-p_y)} \log(p_y) - p_y(1 - p_y). \quad (3.8)$$

To check the impact of weighting factor on gradient updates, consider the first component of the DWB loss,  $L1_{DWB} = -w_y^{(1-p_y)} \log(p_y)$ . It is equivalent to cross entropy loss when  $w = 1$ , for which the loss function gradients are as follows:

$$L_{CE} = -\log(p_y) = -\log\left(\frac{e^{z_y}}{\sum_j e^{z_j}}\right) \quad (3.9)$$

$$\begin{aligned} \nabla_{z_i} L_{CE} &= \nabla_{z_i} \left( -z_y + \log \sum_j e^{z_j} \right) \\ &= \frac{1}{\sum_j e^{z_j}} \nabla_{z_i} \sum_j e^{z_j} - \nabla_{z_i} z_y \\ &= p_i - \nabla_{z_i} z_y \\ &= p_i - \mathbb{1}(y = i), \end{aligned} \quad (3.10)$$

where

$$\mathbb{1}(y = i) = \begin{cases} 1 & ; y = i \\ 0 & ; otherwise \end{cases}$$

When  $w_y \neq 1$ , the gradients of the dynamic weighting factor  $w^{(1-p_y)}$  reduces to:

$$\begin{aligned}
\nabla_{z_i} w_y^{(1-p_y)} &= \nabla_{z_i} w_y \left( 1 - \frac{e^{z_y}}{\sum_j e^{z_j}} \right) \\
&= w_y \left( 1 - \frac{e^{z_y}}{\sum_j e^{z_j}} \right) \log(w_y) \nabla_{z_i} \left( 1 - \frac{e^{z_y}}{\sum_j e^{z_j}} \right) \\
&= -w_y^{(1-p_y)} \log(w_y) [p_y \mathbb{1}(y = i) - p_y p_i].
\end{aligned} \tag{3.11}$$

Using the product rule, we obtain the gradients of  $L1_{DWB}$  as follows:

$$\nabla_{z_i} L1_{DWB} = w_y^{(1-p_y)} [1 - p_y \log(p_y) \log(w_y)] [p_i - \mathbb{1}(y = i)]. \tag{3.12}$$

Thus, when compared with cross entropy loss, the DWB loss weights each data instance by an additional weighting factor. Consequently, the predictions that are less congruent with the provided ground-truth labels are weighed more in the gradient update, which in turn provides more emphasis on neural network training of difficult samples.

## 3.4 Experiments

### 3.4.1 Experimental set-up and Evaluation

*Experiments:* We evaluate the proposed approach on two challenging real-world tasks: Cyber-Intrusion Detection and Skin Lesion Diagnosis, and a detailed description of each is provided in subsequent sections (section 3.4.2 and section 3.4.3). The following loss functions are compared in terms of classification and calibration performance: 1) *Cross entropy* is set as the baseline, 2) *Weighted Cross Entropy* weights each data instance by the inverse frequency, 3) *Focal Loss* down-weights the easy samples, 4) (Proposed) *DWB Loss* dynamically weights loss contribution of each data instance focusing on hard to train instances.

*Classification Evaluation:* In an extreme class imbalanced setting, a classifier that simply predicts any instance as belonging to the majority class could achieve a deceptively high accuracy. We evaluate the model classification performance subject to four different metrics: Precision, Recall/Sensitivity (Detection rate), F-measure and AUROC Score. Let us define a particular class  $j$  as a positive instance and all other classes as negatives. The performance metrics for a particular class

label ( $j$ ) are defined as follows:

$$\begin{aligned}
 Precision_j(Pr) &= TP_j / (TP_j + FP_j) \\
 Recall_j(Re) &= TP_j / (TP_j + FN_j) \\
 F1 - score_j &= (2 \times Pr \times Re) / (Pr + Re),
 \end{aligned}
 \tag{3.13}$$

where  $TP$  are True positives,  $TN$  are True Negatives,  $FP$  are False Positives and  $FN$  are False Negatives.

Precision reflects the proportion of a specific label classified correctly with respect to instances which were predicted to belong in that class. Recall is defined as the proportion of instances that are predicted to belong to a class and truly belong in the class. F1-Score is the weighted harmonic mean of precision and recall. The average of the recall of each class is equivalent to balanced multi-class accuracy. In addition to aforementioned classification metrics, we utilized Area Under the Receiver Operating Characteristics (AUROC) as an evaluation criteria.

*Calibration Metrics:* We evaluate the calibration performance based on Expected Calibration Error (ECE) [64], Maximum Calibration Error (MCE) and Brier Score (BS) [65]. While ECE is the most common calibration metric, it has several drawbacks [66]. We use BS as the primary metric for calibration evaluation which measures the average squared loss between the estimated class membership probabilities and true class value. Lower values indicate better calibration. BS is formally defined as follows:

$$BS = 1/n \sum_{i=1}^n \sum_{j=1}^c (y_{ij} - p_{ij})^2,
 \tag{3.14}$$

where  $n$  is the overall number of instances, with  $y_{ij}$  and  $p_{ij}$  denoting the  $j^{\text{th}}$  element of one-hot encoded class label and predicted probability of the instance  $\mathbf{x}_i$ , respectively.

### 3.4.2 Experiment 1: Cyber Intrusion Detection

An Intrusion Detection System (IDS) dynamically monitors network traffic to efficiently detect cyber-attacks from normal legitimate traffic [67]. As network intrusions represent only a small subset of all network traffic, the size of the benign traffic outweighs that of the malicious traffic. The fact that the overwhelming majority of network traffic will be in the ‘benign’ class and rare positive cases (malicious network traffic) will be in the ‘attack’ class, create an extreme class imbalance problem. Intrusion detection can therefore be interpreted as a multi-class classification problem under high class imbalance.

### 3.4.2.1 Dataset Description

We rely on an intrusion detection data set (CICIDS2017) published by the Canadian Institute for Cyber-Security (CIC) at the University of New Brunswick (UNB) [68]. Captured network flow records in the data set resembles the real-world network traffic and include both normal and malicious attack traces. This flow-based data is captured within a five-day timeframe in 2017 and contains 3.1 million flow records. Each network flow record is characterized by 86 features which can be categorized into *time-based features* (e.g. flow duration and inter-arrival packet time), *size of payload data* (e.g. total application bytes and maximum size of the packets) and *packet count* (e.g. source to destination packet count). Certain attack classes in the data set are highly underrepresented categorizing intrusion detection for CICIDIS2017 as an extremely imbalanced multi-class classification problem.

### 3.4.2.2 Implementation (Intrusion Detection System Model Overview)

While deep learning models can extract features automatically, conventional machine learning classifiers involve a feature selection phase and hence, implemented under three stages: (a) *Pre-processing phase*: includes data cleaning, stratified Train-Validation-Test split procedure and data transformations; (b) *Feature Selection phase*: Implementation of correlation analysis followed by feature selection through Recursive Feature Elimination with Cross Validation (RFE-CV); (c) *Classification phase*: involves model fitting and performance evaluation. Classification performance of conventional classifiers and deep neural networks with different loss functions is then compared.

*Data Pre-Processing Phase*: The data set is comprised of separate attack files for each attack class. We first combined all attack records into Denial of Service (DoS) attack file which encompasses the largest number of Benign records. However, the prevalence rate of each attack in individual data files remain approximately the same after merging them. Two attack types (Heartbleed and Infiltration) were omitted since they constitute only a very small fraction of flow records. Individual web attack classes were merged together into a single web attack category. Nominal Features that are related to a specific network and another ten features that contained all zero entries were removed from the data frame. After the pre-processing stage the data-frame dimension reduces to 911421 records with 66 network flow feature variables. For training purposes, we only considered 10% of the data (stratified sample). The network activity flow distribution across different attack categories after pre-processing stage is depicted in Table 3.1 and Figure 3.3. Three subsets were

Table 3.1: The distribution of network flows in each attack category

Class Category	Count	Percentage(%)
Benign	44002	48.2780
DoS Hulk	23107	25.3525
PortScan	15893	17.4374
DDoS	4183	4.5895
DoS GoldenEye	1029	1.1290
FTP-Patator	794	0.8712
SSH-Patator	590	0.6473
DoS slowloris	580	0.6364
DoS Slowhttptest	550	0.6034
Web Attack	218	0.2392
Bot	197	0.2161

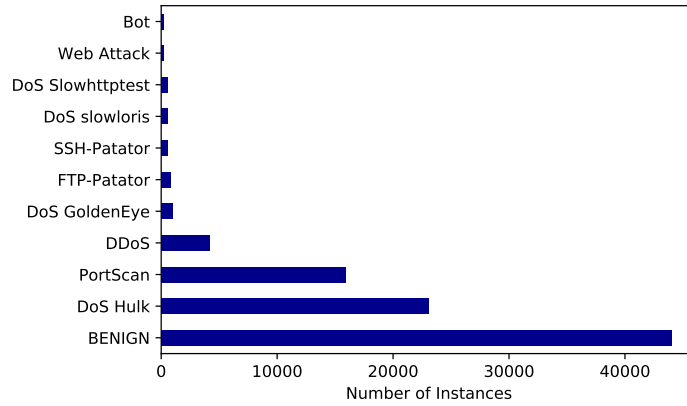


Figure 3.3: Network activity flow distribution with network flow-count varying sharply across different attack categories.

obtained in a stratified manner for training (60%), validation (20%) and testing (20%) purposes. Stratification enables to randomly split the data set while retaining the correct class distribution in each subset, which is the recommended way of splitting data under class imbalance [69]. In order to avoid biased results, feature vector is transformed by scaling each feature to a [0,1] range. Categorical variable ‘class label’ was transformed through one-hot encoding.

*Feature Selection Phase for conventional classifiers:* For conventional Machine Learning (ML) classifiers, we conducted a feature selection procedure to identify representative and distinguishable features for intrusion detection. We first conducted a correlation analysis to identify possible correlations. Considering 0.90 as the correlation coefficient threshold, 32 features with a correlation magnitude greater than 0.90 were removed. Then, an optimal subset of features was obtained through Recursive Feature Elimination with Cross-Validation (RFE-CV) which is used to train



classical ML classifiers. The selected subset containing 11 features includes the total number of data packets in the forward direction, the total quantity of bytes in the forward direction, the maximum and mean values of the packet’s length in bytes in the forward direction, the maximum value in bytes of the packet’s length in the backward direction, the mean and standard deviation of the inter-arrival time of the flow in both directions, the number of packets per second in the backward direction, the minimum length of the packets registered in the flow in both directions, the total number of bytes sent in the initial window in the forward and backward directions.

*1D Convolutional Neural Net (1D-CNN) Model:* The one-dimensional convolution neural network based intrusion detector had the best performance compared with other DL models. The implementation of cost sensitive classification with CNN does not require a feature selection phase since convolution layers are capable of extracting better representations from data automatically. We consider 1D-CNN with convolutions in the spatial domain. The applicability of the 1D-CNN model in CICIDS2017 network flow data can be justified as follows: We notice that there is high correlation among features and data contain features that belong to similar groupings. Thus, there is a local pattern in the features and the relative spatial positioning of the data is relevant with local relationships in data providing more predictive information for the classification task. Hence, the idea of local spatial correlation in CNN translates well to the problem at hand. We expect interesting features to depend on short consecutive sub-sequences of the input. We treat the input features as spatial dimension and the kernel is convolved over input features. We expect 1D-CNN model to capture specific patterns from successive input features and thus derive a more robust representation of features which contain important information for identification of malicious network flows.

The intrusion detector 1D-CNN model architecture is depicted in Figure 3.4. It involves an input layer (shape 66 x 1) , two convolutional layers with one-dimensional filter kernels of size 3, max pooling layer with sub-sampling factor 2, a flattening layer, one dense layer and a final output layer with the number of nodes equal to number of classes. The activation function of the hidden dense layers is Rectified Linear Unit (ReLU) and Softmax is employed in the output layer for the multi-class classification. Each network is trained for 200 epochs with Adam optimization [70] method.

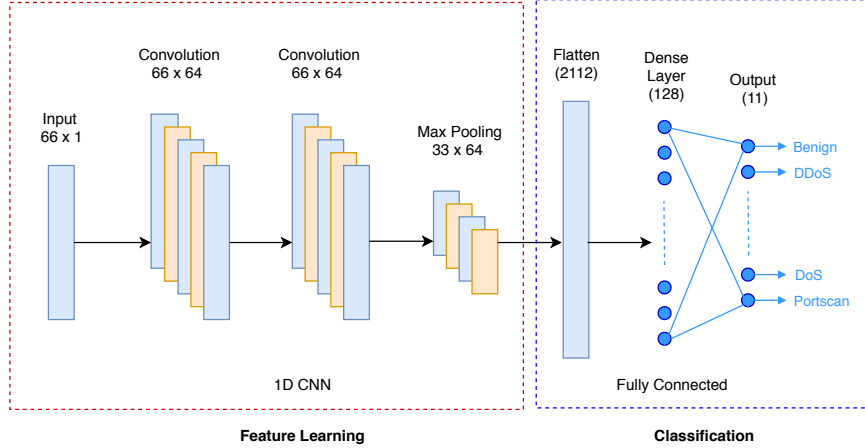


Figure 3.4.: 1D-CNN Model Architecture. It includes two convolution layers and a pooling layer followed by a standard fully connected neural network.

### 3.4.2.3 Experimental Results

Using the reduced feature subset obtained in the feature selection stage, we tested several widely used traditional machine learning classifiers including Multinomial Logistic Regression, Random Forests, Decision Tree, Gradient boosting and XGBoost. Their performance in terms of average precision, recall and F1-Score is presented in Table 3.2 with multinomial logistic regression having the worst performance. This result is not surprising and is consistent with the previous research which have proven the performance degradation of conventional logistic regression under class imbalance [60]. Except for multinomial logistic regression which is highly affected by the imbalanced class distribution, other conventional classification algorithms seem to be performing well. However, their performance is comparatively low in comparison to DL Models. While we experimented with several different DNN and 1D-CNN model architectures, we only included the results of the best performing 1D-CNN model in the paper. The average classification results in Table 3.2, as well as the class-wise classification performance of 1D-CNN Model trained with different loss functions in Table 3.3 suggest that the proposed DWB loss clearly outperforms the other commonly used objective functions in cost sensitive learning. Specifically, F1 score and recall or the ‘Detection Rate’ of attacks is highest with the proposed method for the most extremely imbalanced classes, such as Bot attacks which occupy only 0.2% of data.

For CICIDS2017 data we provide only the results of our primary calibration metric, Brier Score since values for other calibration metrics are extremely small that the difference is insignificant

Table 3.2: CICIDS2017 Dataset: Average metric values (Percentages)

classification Algorithm	Precision	Recall	F1-score	AUROC Score
<b>Conventional ML Classifiers</b>				
Multinomial Logistic Regression	37.75	22.26	24.67	58.89
Decision Tree	94.43	95.75	95.05	97.85
Random Forest	95.57	95.66	95.60	97.80
XGboost	95.90	95.31	95.51	97.61
Gradient Boosting	91.55	91.36	91.02	95.64
<b>DL: 1D CNN Model</b>				
CE Loss	97.15	96.00	96.50	97.97
Weighted CE	97.38	97.44	97.40	98.69
Focal Loss	96.96	98.05	97.49	98.89
DWB Loss	<b>97.52</b>	98.00	<b>97.74</b>	<b>98.99</b>

Table 3.3: CICIDS2017 Dataset: Class-wise Classification Performance

		Benign	Bot	DDoS	DoS GoldenEye	DoS Hulk	DoS Slowhttptest	DoS Slowloris	FTP Patator	PortScan	SSH Patator	Web Attack
Precision	CE Loss	99.49	84.38	100	98.54	99.65	95.5	99.13	99.37	100	99.14	93.48
	Weighted CE	99.87	85.71	99.76	97.15	99.14	96.43	99.13	99.37	100	96.61	93.33
	Focal Loss	99.87	85.71	99.76	97.14	99.14	96.43	99.13	99.37	100	96.61	93.34
	DWB Loss	99.82	88.1	99.88	99.5	99.5	95.5	99.13	100	100	100	91.3
Recall	CE Loss	99.67	69.23	99.88	98.54	99.5	96.36	98.28	99.37	99.97	97.46	97.72
	Weighted CE	99.31	92.31	100	99.03	100	98.18	98.28	99.37	99.96	96.61	95.45
	Focal Loss	99.31	92.31	100	99.03	100	98.18	98.28	99.37	99.97	96.61	95.45
	DWB Loss	99.61	94.87	99.88	97.57	100	96.36	98.28	99.37	99.96	96.61	95.45
F1-Score	CE Loss	99.58	76.06	99.94	98.54	99.58	95.93	98.7	99.37	99.98	98.29	95.56
	Weighted CE	99.67	91.14	99.7	97.8	99.8	96.83	98.7	98.4	99.98	98.29	91.11
	Focal Loss	99.59	88.88	99.88	98.08	99.57	97.3	98.7	99.37	99.98	96.61	94.38
	DWB Loss	<b>99.72</b>	<b>91.36</b>	99.88	<b>98.54</b>	99.75	95.93	<b>98.7</b>	<b>99.68</b>	<b>99.98</b>	<b>98.29</b>	93.33

(Table 3.4). The Brier Score is at its lowest when trained with the proposed DWB loss function implying better calibration.

Table 3.4: CICIDS2017 Dataset: Calibration Performance

	CE Loss	Weighted CE	Focal Loss	DWB Loss
Brier Score	0.0067	0.0065	0.0116	<b>0.0056</b>

### 3.4.3 Experiment 2: Skin Lesion Diagnosis

Skin lesions are among the most common cancers worldwide with over 5,000,000 newly identified cases in the United States every year. Melanoma is the deadliest skin malignancy, but if diagnosed early has a survival rate which exceeds 95%. To facilitate early and accurate detection

of skin cancers, a fast and automated diagnosis system is crucial. Dermoscopy is a skin imaging modality which is pivotal in detection of skin malignancies and supports towards implementation of automated algorithmic systems. Lesion detection is one of the most challenging tasks in medical imaging due to high similarity between lesions and intra-class variations with respect to texture, color, size, shape and location. Class imbalanced nature of the diagnosis task makes it even more challenging.

### 3.4.3.1 Dataset Description

We utilize ISIC2019 challenge skin lesion data [71–73], published by International Skin Imaging Collaboration (ISIC) in ISIC Archive which is the largest publicly available repository of dermoscopic images. The goal is to classify skin lesions based on 25,3331 dermoscopy images available for training which are unequally distributed among 8 different lesion categories. A sample of skin lesion images in the ISIC2019 data set is provided in Figure 3.5. The skin lesion diagnosis distribution is presented in Table 3.5 and Figure 3.6.

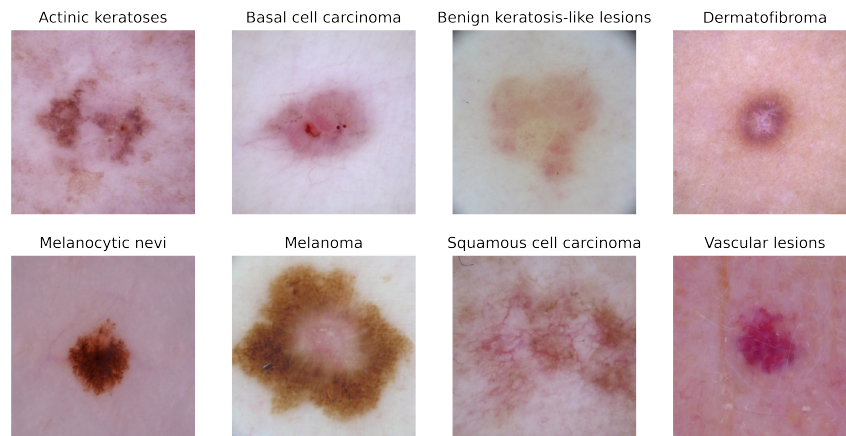


Figure 3.5.: A sample of different skin lesion categories from the ISIC2019 data set.

### 3.4.3.2 Implementation Details

To ensure the class distribution remains the same, we split ISIC2019 data to train-test-validation subsets in a stratified manner such that train data contains 19,173 data entries, with validation and test sets each having 1070 unique entries. Both skin lesion dermoscopic image data and meta data were employed for lesion detection model implementation following a dual input strategy. Meta data contains patient age, gender and anatomy site. Meta and dermoscopy data were pre-processed

Table 3.5: The distribution of skin lesion diagnostic category

Diagnosis Category	Count	Percentage(%)
Melanocytic nevi (NV)	12875	50.83
Melanoma (MEL)	4522	17.85
Basal cell carcinoma (BCC)	3323	13.12
Benign keratosis (BKL)	2624	10.36
Actinic keratosis (AK)	867	3.42
Squamous cell carcinoma (SCC)	628	2.48
Vascular lesion (VASC)	253	1.00
Dermatofibroma (DF)	239	0.94

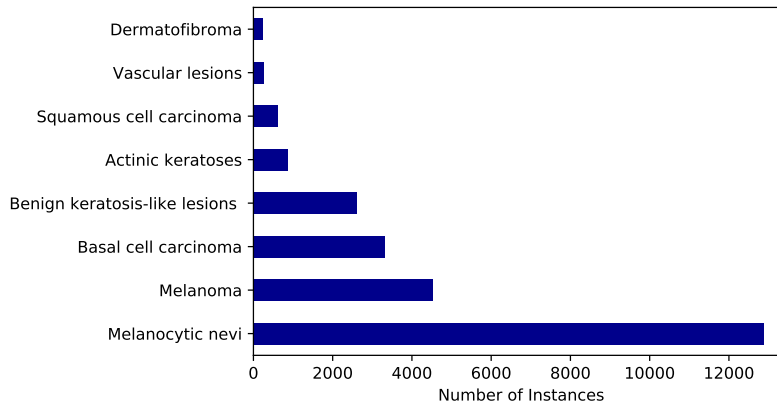


Figure 3.6.: Skin Lesion Diagnosis distribution with lesion count varying sharply across different diagnosis categories.

prior to training and images were augmented during training with random flipping, color, shift and rotation transformations to ensure robustness to deformations, thereby better generalization.

*Model Architecture and Training:* We relied on established methods for computer vision and used the state-of-the-art deep learning models for image classification. We incorporated an EfficientNet architecture (Figure 3.7), specifically EfficientNet (EN-B3), [74] since it performed significantly better than the other experimented models. To be consistent with the chosen ImageNet pre-trained model, the input images were resized to 300 x 300. The schematic diagram of the dual input neural network model architecture is depicted in Figure 3.8. Network was trained for 30 epochs with Stochastic Gradient Descent (SGD) [75] optimization algorithm.

### 3.4.3.3 Experimental Results

We evaluate the impact of different loss functions for training to diagnose skin lesions and the result of this analysis is presented in Table 3.6 and Table 3.7. Average classification metric val-

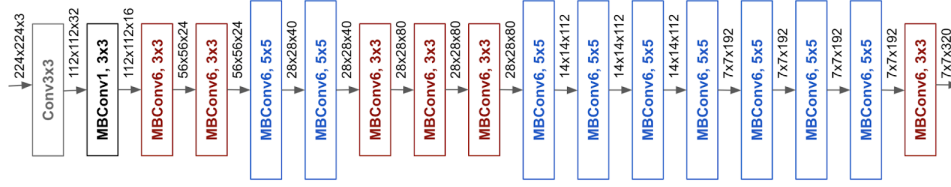


Figure 3.7.: The architecture of EfficientNet-B0 [76]. The main building block of EfficientNet is mobile inverted bottleneck convolution(MBConv).

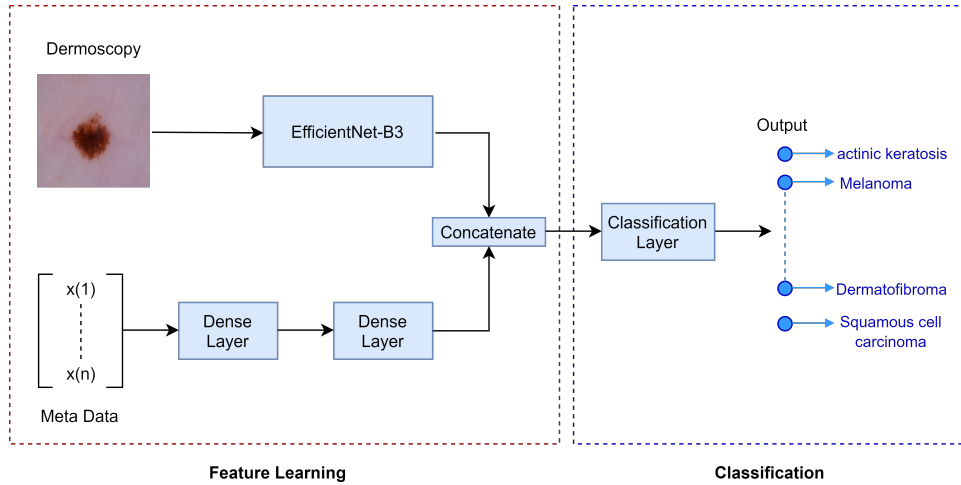


Figure 3.8.: Schematic diagram of the dual-input neural network model architecture composed of a 2D-CNN (EfficientNet-B3) and fully connected model.

ues of diagnosis categories and class-wise classification performance suggest that the DWB loss considerably outperforms the other loss functions in terms of classification.

Table 3.6: ISIC2019 Dataset: Average Metric Values (Percentages)

Loss Function	Precision	Recall	F1-Score	AUROC
CE Loss	66	64	65	80
Weighted CE	67	66	66	81
Focal Loss	64	60	61	78
DWB Loss	<b>69</b>	<b>66</b>	<b>67</b>	<b>82</b>

Calibration Performance evaluation results for ISIC2019 data are presented in Table 3.8 and Figure 3.9. With DWB objective function, we observe calibration results which are much improved over the other losses trained with the same network.

The experimental results are in consistent across detection tasks in different domains, implying that when trained with the proposed loss function, the model surpasses the performance of conventional classifiers in terms of both classification and calibration.

Table 3.7: ISIC2019 Dataset: class-wise classification (Percentages)

		Skin Lesion Diagnosis Category							
		AK	BCC	BKL	DF	MEL	NV	SCC	VASC
Precision	CE Loss	50	77	70	50	49	92	64	78
	Weighted CE	41	79	71	75	49	93	50	78
	Focal Loss	36	72	71	62	51	91	50	75
	DWB Loss	42	78	74	88	56	93	50	75
Recall	CE Loss	62	81	63	50	54	93	41	70
	Weighted CE	56	76	67	75	53	93	36	70
	Focal Loss	50	67	67	62	49	93	27	60
	DWB Loss	46	79	70	88	60	93	35	67
F1-Score	CE Loss	56	79	67	50	51	93	50	74
	Weighted CE	47	77	69	75	51	93	42	74
	Focal Loss	42	70	69	62	50	92	35	67
	DWB Loss	46	<b>79</b>	<b>70</b>	<b>88</b>	<b>60</b>	<b>93</b>	35	67

Table 3.8: ISIC2019 Dataset: Calibration Metrics

Loss Function	ECE	MCE	Brier Score
CE Loss	0.0553	0.2212	0.2596
Weighted CE	0.0330	0.1321	0.2622
Focal Loss	0.0612	0.2454	0.2458
DWB Loss	<b>0.0295</b>	<b>0.0938</b>	<b>0.2389</b>

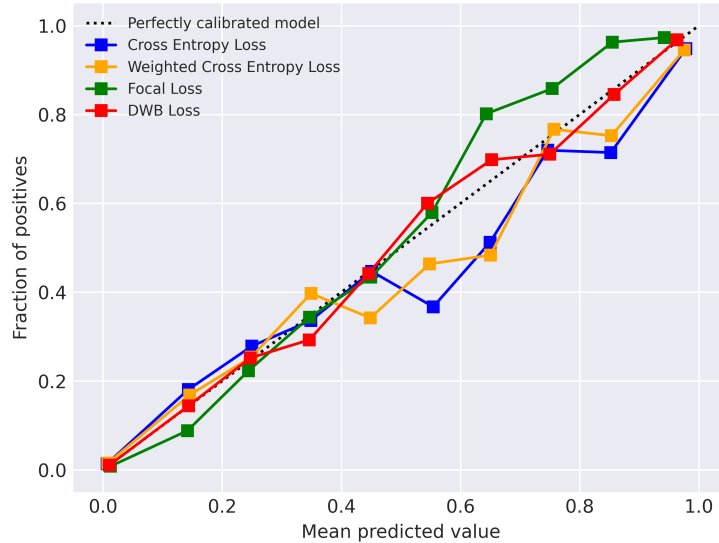


Figure 3.9.: Probability calibration plot for ISIC2019 data. From the calibration curve, it is clearly evident that models trained with standard cross entropy loss and its fixed weighed counterpart is poorly calibrated. The plot further confirms that the proposed DWB loss is classification calibrated.

### 3.5 Contributions and Concluding Remarks

The study leads to the following conclusions and contributions:

- To address class imbalance encountered in many practical, real-world classification tasks, we presented a self-adapting weighting approach and introduced a novel loss function, named Dynamically Weighted Balanced (DWB) Loss. Weighting scheme is based on class frequency of training data and prediction difficulty of individual data instances. The prediction difficulty is determined by the prediction score produced by the neural network.
- We further demonstrated that the regularization component in the proposed loss function leads to improved calibration performance.
- Experiments in different domains: cyber intrusion detection (tabular data) and skin lesion diagnosis (image classification) show consistent results implying robust generalization. A considerable performance improvement was observed in rare minority classes with the proposed DWB loss function over different kinds of other widely adopted loss functions when tested for the same model architecture.
- Presented method can be adapted for any classification or segmentation task owning broad applicability and its superior performance suggests the potential of cost-sensitive deep learning based models for real-life deployment.



**CHAPTER 4**  
**BAYESIAN-BASED PROBABILISTIC DEEP LEARNING FOR UNCERTAINTY**  
**QUANTIFICATION WITH APPLICATIONS IN BIO-MEDICAL IMAGE**  
**SEGMENTATION**

**4.1 Introduction**

Technical advances have led multiple innovations into clinical practice supporting clinicians with assistive information for clinical decision making. Multiple efforts have been made to integrate deep learning (DL) and image-based findings in diagnostics, subsequently augmenting the efficacy of clinical care. While there is a significant increase in clinical research on deep learning-based models in healthcare, the level of uncertainty in model predictions is often inaccessible. Reliable uncertainty estimation is imperative in safety-critical applications in medicine such as in radiation therapy planning in cancer care, where radiation should be avoided in healthy sub-cortical structures.

This study aims to analyze segmentation uncertainties in neuroimaging, specifically, in brain tumor diagnostics, which is an area of active research. Brain tumors are among the deadliest malignancies. Gliomas are the most predominant primary tumors that develop within the brain parenchyma. With a degree of biological aggressiveness spanning from slower-growing low-grade gliomas (LGG) to more rapidly progressive high-grade gliomas (HGG), varying biological properties, patient prognosis, and treatment strategies, gliomas are highly heterogeneous. In tumor cell morphology studies, neuroimaging plays a crucial role. Magnetic resonance imaging (MRI) as the most widely adopted neuroimaging modality, serves as a non-invasive technique in lesion detection and segmentation. MRI-guided diagnosis facilitates treatment which entails surgical tumor resection, radiation, and chemotherapy. In an automated system pipeline of tumor analysis, developing robust volumetric automated segmentation algorithms is of utmost importance for extracting tumor morphological information and to ascertain the degree of surgical resectability. Segmentation enables volumetric tumor delineation, progression evaluation, and the quantitative assessment of tumor diagnostic features. Segmentation of glioma entails partitioning of tumor cells into histolog-

ical sub-regions as peritumoral edema (ED), necrotic core (NCR), enhancing and non-enhancing tumor core (ET/NET), properties of which are reflected in the intensity profiles in tumor scans generated by MRI. Different types of imaging modalities provide useful biological information related to tumor-induced neural tissue changes. The T1-weighted (T1), T1-weighted contrast-enhanced (T1ce), T2-weighted (T2), and T2 fluid-attenuated inversion recovery (FLAIR) modalities are the routinely used brain MRI image sequences. A 2D slice of each modality and the segmented scan is depicted in Figure 4.1. While segmentation aids in precise localization and diagnosis through objective quantification of tumor tissue composition and size, proper segmentation remains a key challenge due to the intra-tumoral heterogeneity of tumor cells.

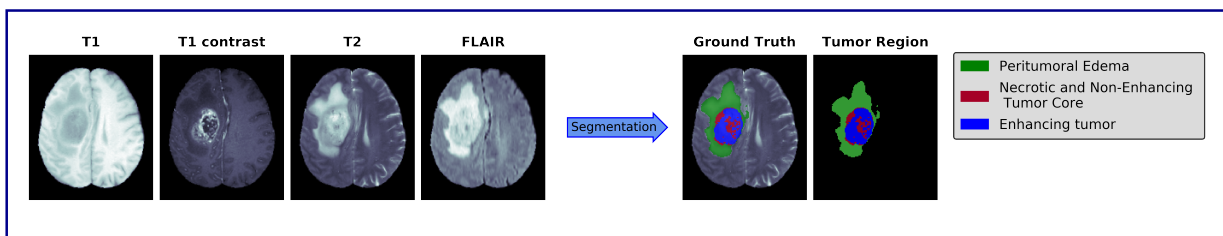


Figure 4.1.: On the left is an example of glioblastoma brain tumor in T1, T1-contrast, T2 and FLAIR modalities. On the right is the ground truth segmentation of the tumor which specifies the class of each voxel [1].

Since lesion outlines are only defined by variations in intensity compared to surrounding non-cancerous tissues, even manual segmentation by radiologists demonstrates considerable variation. The intrinsically heterogeneous nature of tumors (in appearance, shape, and histology) poses an additional layer of complexity. Thus, despite substantial advances made towards understanding the molecular pathology of gliomas, challenges remain in the development of robust automated segmentation models. Replacing deterministic model predictions with probability distributions would allow characterizing model reliability, thereby enabling clinicians to review and revise unreliable model predictions, which would have enormous potential value for clinical adoption.

While there are many forms of uncertainty relating to the measurement noise, structure of the DL model (choice of architecture, number of hidden units and layers, etc.) and model parameters, uncertainty is broadly categorized as epistemic and aleatoric (Figure 4.2), which explains uncertainty in the model and the inherent noise in observations due to intrinsic stochasticity of the system, respectively [77]. The combination of the two uncertainties resulting from model and data forms the predictive uncertainty [78]. Integrating uncertainty information allows to account for model misspecifications and evaluate the robustness for domain shift when modeling out-of-distribution

samples. While uncertainty does not disappear in large data sets, epistemic uncertainty diminishes as more data becomes available. However, aleatoric uncertainty resulting from intrinsic randomness of data is irreducible with more data [79], but can be reduced by increasing the measurement precision. The majority of the studies typically address the uncertainty arising from one single source where most studies are geared towards parametric uncertainty and total uncertainty remains briefly addressed in the past literature.

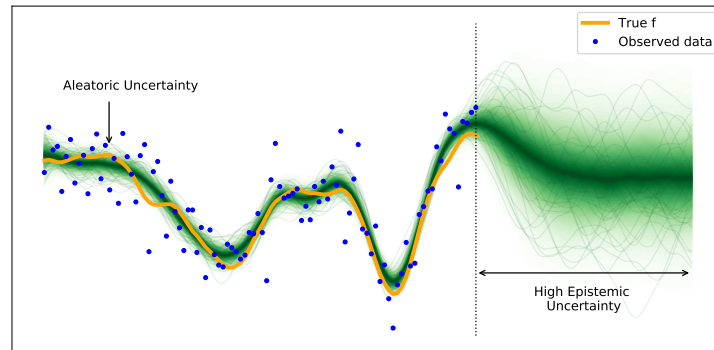


Figure 4.2.: Epistemic uncertainty emerges from lack of knowledge resulting from a priori unknown data patterns and represents the uncertainty in model parameters that best explains a given data set. Aleatoric uncertainty represents randomness inherent in observations.

This study aims to develop a unified procedure for uncertainty quantification in biomedical image segmentation integrating Bayesian inference with deep learning. Strengths and weakness of the formulated framework is subsequently analyzed using complex neuroimaging data (BRATS2020). The developed fully automated system for 3D multi-modal MRI brain tumor segmentation empowered with uncertainty estimation demonstrates the proposed approach is a promising candidate for quantifying uncertainty in biomedical image analysis.

## 4.2 Related Work

Driven by the advances in computer vision, an increasing number of studies in clinical research have examined deep learning based algorithmic solutions for automated segmentation of medical images [80–82]. However, the research in analyzing segmentation uncertainty in biomedical imaging remains limited, which is nevertheless crucial for successful clinical translation. Here we present a concise overview on novel contributions towards uncertainty estimation in medical imaging.

While Bayesian principles offer a theoretical framework for quantifying uncertainty, the computational cost of Bayesian Deep Nets can be prohibitively expensive. Hence, research efforts have been

made to characterize uncertainty via approximate Bayesian inference, primarily, sampling-based Monte Carlo methods and optimization-based variational inference. Among the well-established methods for uncertainty estimation are drop-out at test-time [14] and Deep Ensembles [15]. Owing to its simplicity and scalability, Monte Carlo dropout based uncertainty has been leveraged in a considerable body of literature in medical domain, for instance, in pulmonary nodule detection [83], diabetic retinopathy diagnosis [84], brain segmentation [85], phenotype prediction [86], ischemic stroke lesion segmentation [87], and multiple sclerosis lesion segmentation [88]. Similarly, researchers have also explored uncertainty as quantified by test-time augmentation [89]. A closer look to the literature on uncertainty, however, reveals a number of shortcomings in popular methods. As argued in [90], MCMC dropout approach estimates the risk or intrinsic stochasticity in a model, rather than the model uncertainty. Moreover, both deep ensembles and dropout involve averaging over several models, which can be computationally expensive for very large networks.

More recent work have explored modeling the conditional probability distribution over the label maps given an image, allowing for multiple plausible hypotheses for a single image [91–93]. For instance, in [92] authors propose a Probabilistic Hierarchical Segmentation (PHiSeg) approach where they adopt a variational auto-encoder framework and use separate latent variables at each level of the network. However, these hierarchical architectures are memory intensive and involve high-computational complexity.

### 4.3 Segmentation model construction from a probabilistic perspective

#### 4.3.1 Parameter Uncertainty and Probabilistic layers

Let a training set  $\mathcal{D}$  with  $n$  samples be denoted by  $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ , where  $\mathbf{x}_i \in \mathbb{R}^d$  is the  $d$ -dimensional feature vector and  $y_i$  is the corresponding output. In the context of voxel-wise classification, each input  $x$  is an intensity and  $y \in \{1, \dots, K\}$  is the space of  $K$  possible class labels. In a standard neural network-based segmentation task, the network learns a function  $f: \mathcal{X} \rightarrow \mathcal{Y}$  which maps images  $\mathbf{x} \in \mathcal{X}$  to a segmentation map containing voxel-wise target labels  $y \in \mathcal{Y}$ . The objective is to learn a mapping that maximizes the conditional probability  $p(y|x)$ .

The Bayesian framework enables capturing the uncertainty over model weights (parameter uncertainty) that results in probabilistic interpretations of model predictions. Therefore, we seek to learn a probabilistic function  $f: \mathcal{X} \rightarrow \mathcal{Y}$  by placing Bayesian weights over the network layers during training. Previous studies [94] have emphasized that Bayesian model average performance

is robust to the choice of prior over the weights and, hence we assume a standard Gaussian prior,  $W \sim \mathcal{N}(0, \mathbb{I})$ . We are particularly interested in the predictive distribution stated in Eq. (2.12):

$$p(y^*|x^*, \mathcal{D}) = \int_{\Omega} p(y^*|x^*, \omega) p(\omega|\mathcal{D}) d\omega, \quad (4.1)$$

where  $p(y^*|x^*, \omega)$  is the conditional predictive distribution and  $p(\omega|\mathcal{D})$  is the posterior over parameters.

The posterior computation is analytically intractable and traditional MCMC methods do not scale better over highly parametrized deep neural networks and large data sets. Thus, for computational reasons, we adopt the variational inference [10, 95] approach (Figure 4.3 and Algorithm 2), where posterior over weights is approximated by a tractable family of distributions  $q_{\theta}(\omega)$ , parameterized by  $\theta$ . Typically, the reverse KL-divergence  $KL[q_{\theta}(\omega)||p(\omega|\mathcal{D})]$  is minimized to fit the approximation. Since it includes the intractable posterior, it is rearranged to maximize the evidence lower-bound (ELBO):

$$\min_{\theta} KL[q_{\theta}(\omega)||p(\omega|\mathcal{D})] = \max_{\theta} \mathbb{E}_{q_{\theta}(\omega)}[\log p(\mathcal{D}|\omega)] + KL[q_{\theta}(\omega)||p(\omega)], \quad (4.2)$$

where  $p(\omega)$  denotes the prior on  $\omega$  and  $\mathbb{E}_{q_{\theta}(\omega)}[\log p(\mathcal{D}|\omega)]$  is the expected log-likelihood such that  $p(\mathcal{D}|\omega) = \prod_{i=1}^n p(y_i|x_i, \omega)$ .

At the inference time, We obtain the approximate predictive distribution under the mean field variational bayes (MFVB) assumption:

$$p(y^*|x^*, \mathcal{D}) = \int_{\Omega} p(y^*|x^*, \omega) q_{\theta^*}(\omega) d\omega, \quad (4.3)$$

where  $q_{\theta^*}(\omega)$  is approximated by a product of independent Gaussians:

$$q_{\theta}(\omega) = \prod_{i=1}^{|\omega|} \mathcal{N}(\omega_i|\mu_i, \sigma_i^2), \quad (4.4)$$

where  $\theta = \{\mu_i, \sigma_i^2\}$  are the variational parameters. The elements  $\sigma_i$ 's are from a diagonal covariance matrix, thus implying that the neural network weights are considered to be uncorrelated.

*Justification for MFVB:* Mean field assumption posits isotropic Gaussian priors over weights assuming neural network weights are independent. While prior studies have experimented with more expressive families of distributions such as a multivariate Gaussians, incorporation of a-posteriori

correlations incur additional parameter overhead imposing a heavy burden on the inference model. Thus, their applicability is limited in practice due to computational inefficiency. As argued in [96], complex posterior approximations for weights are not necessary in deep bayesian networks since the difference between the full-covariance and mean-field approximations to the true posterior reduces as the network becomes deeper.

*Network Training:* An iteration in the training process constitutes a forward and a backward pass. During forward pass, the approximate cost function in Eq. 4.2 is evaluated through a single sample drawn from the variational posterior distribution. Thus, in the stochastic forward pass, a sampling step is involved and therefore, in order to implement backpropagation, we need to incorporate the re-parameterization trick. More specifically, a function is defined as  $t(\mu, \sigma, \epsilon) = \mu + \sigma \odot \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, 1)$ ,  $\mu$  is the mean,  $\sigma$  is the standard deviation and  $\odot$  is the element-wise multiplication, which essentially defines a deterministic function from which the gradients of  $\mu$  and  $\sigma$  can be calculated.

At inference, an unbiased estimator for expectation of posterior predictive distribution can be obtained by sampling from the approximate distribution  $q_\theta(\omega|\mathcal{D})$ :

$$\mathbb{E}_q[p(y^*|x^*, \mathcal{D})] = \int p(y^*|x^*, \omega) q_{\theta^*}(\omega) d\omega \approx \frac{1}{T} \sum_{t=1}^T p_{\omega_t}(y^*|x^*), \quad (4.5)$$

where  $T$  represents the number of samples from multiple forward passes.

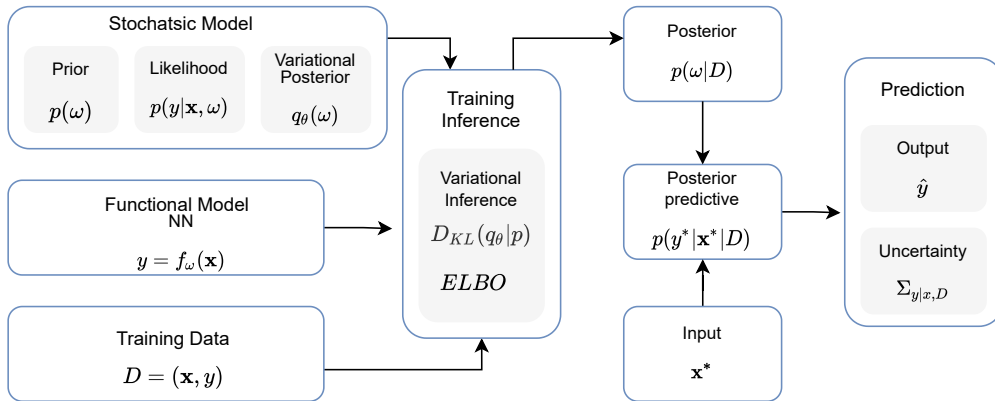


Figure 4.3.: Process overview for developing a deep probabilistic neural network via variational inference.

### 4.3.2 Data-dependent Uncertainty

Data uncertainty is typically modeled as part of the likelihood function. In regression, for instance, a distribution such as a Gaussian can be placed over the logits in order to induce a noise corruption process. This allows us to obtain data dependent uncertainty in the form of variance of the noise. In classification, network output can be treated as a distribution as described below.

In a multi-label setting, k-class discriminative task is transformed into  $k$  independent binary classes. Let the conditional distribution in binary classification be denoted by  $g(x) := p_{\hat{Y}|X}(1|x)$ . Note that in practice, for binary classification tasks we typically adopt a soft classifier  $g: \mathcal{X} \rightarrow [0, 1]$ , which can be obtained through sigmoid normalization. Let affine-transformed logits be denoted by  $z_j = w_j^T f_k(x) + b_j$ , where  $f_k(x)$  is the model output in the penultimate layer, with  $w_j$  and  $b_j$  representing the weights and biases in the last layer  $j$ . Then the sigmoid function for each specific class is represented by:

$$p(\hat{y}^{(k)}|x) = \frac{1}{1 + \exp(-(w_k^T f_\omega(x) + b_k))}. \quad (4.6)$$

Since we consider k-class binary classification, the output is the probability that the label is one, and hence each label can be interpreted as a Bernoulli random variable:  $Y \sim Ber(p)$  where  $p = \sigma(z)$ , such that  $\sigma$  is the sigmoid function. Therefore, the last layer can be treated as a probabilistic layer that parameterizes a Bernoulli distribution. Input-dependent uncertainty exists in all classification tasks, however, aleatoric uncertainty is closely related to the calibration, which implies how closely the predicted probability reflects the ground truth likelihood.

To this end, we incorporate a classification-calibrated weighted binary loss in the objective function. In the k-class discriminative task, we can compute the cross-entropy loss component as the summation of multiple binary cross entropy terms. Adapting from the DWB loss defined in chapter 3 that proved to improve calibration, we define a (fixed) weighted binary cross entropy as follows.

$$\mathcal{L}_{WBCE} = -w y \log(p) + (1 - y) \log(1 - p) + 2p(1 - p), \quad (4.7)$$

where setting  $\beta > 1$  reduces the number of false positives,  $y$  is the true label and  $p$  is the predicted label.

We also leverage the DICE loss component in the loss function which is good at discriminating between the foreground pixels from the background pixels, thus addressing class imbalance:

$$\mathcal{L}_{DICE} = - \sum_{c=0}^2 \frac{\sum_{i,j,k=0}^{127} Y_{cijk} \hat{Y}_{cijk}}{\sum_{i,j,k=0}^{127} Y_{cijk} + \sum_{i,j,k=0}^{127} \hat{Y}_{cijk}}. \quad (4.8)$$

Recall that the variational objective is to minimize the ELBO defined in 4.2. This leads to a combined loss function which is the sum of modified cross entropy, DICE loss and the ELBO term. We thus define our version of combined loss as follows:

$$\mathcal{L}_{combined} = \mathcal{L}_{WBCE} + \mathcal{L}_{DICE} + \mathcal{L}_{ELBO}, \quad (4.9)$$

where the contribution from each loss is equally weighted.

The training procedure of the Deep Probabilistic Learning model can then be summarized as follows:

---

**Algorithm 2** Variational Inference-based Deep Probabilistic Learning

---

**Input:** Data set  $D : \{x_i, y_i\}_{i=1}^n$ , Approximation of the posterior distribution  $q_{\theta}(\omega)$

**Output:** posterior distribution of the network parameters

- 1: **while** not converged **do**
  - 2:   Sample minibatch:  $D^* : \{x_i, y_i\}_{i=1}^m$
  - 3:   Sample from  $q_{\theta}(\omega)$  the variational parameters  $\theta = (\mu_i, \sigma_i^2)$  for each weight of the network:  
 $\omega^{*(i)} \sim q_{\theta_i}(\omega^{(i)})$
  - 4:   Using the reparameterization trick, parameterize the network with the sampled parameters
  - 5:   Forward pass with batch of data
  - 6:   Compute stochastic gradients of the objective by backpropagation
  - 7:   Update parameters:  $\theta = \theta + \alpha \nabla_{\theta} \mathcal{L}$
  - 8: **end**
- 

### 4.3.3 Uncertainty Estimation

In segmentation, prediction map is obtained via voxel-wise classification and hence uncertainty estimates are also produced voxel-wise, resulting in uncertainty maps with dimensions same as that of the original input image. We measure each source of uncertainty as follows:

*Data Uncertainty:* Given a test input  $x^*$  and training data  $D$ , the expected data uncertainty can be captured by  $\mathbb{E}_{p(\omega|D)} \mathbb{H}[p(y|x^*, D)]$  where  $\mathbb{H}[p(y|x^*, D)]$  is the entropy of the model’s posterior over classes calculated as:

$$\mathbb{H}[p(y|x^*, D)] = - \sum_c p(y = c|x^*, \omega) \log p(y = c|x^*, \omega), \quad (4.10)$$



where  $y$  is the segmentation label,  $c$  is the segmentation category,  $p(y = c|x^*, w)$  is the network output which is the probability of input  $x^*$  being in class  $c$ , and  $w$  represents the model parameters.

*Model uncertainty:* Prediction is done through multiple forward passes. The final prediction is the mean prediction (distribution mean) and the model uncertainty is reflected through the variability in the mean prediction. Let  $Y^i = (y_1)^i, \dots, (y_M)^i$  represents the predicted labels of voxel  $i$  for each iteration  $M$ . Then the predictive variance can be computed as follows:

$$V_{q_\theta(\omega)}[\mathbb{E}_{p(y|\omega, x)}[y]] = \frac{1}{M} \sum_{m=1}^M (y_m^i - y_{mean}^i)^2, \quad (4.11)$$

where  $M$  is the number of iterations.

*Total uncertainty:* Predictive uncertainty encapsulates both parameter and data uncertainty. We quantify total uncertainty through the entropy of mean predictions,  $\mathbb{H}[\mathbb{E}_{p(\omega|\mathcal{D})}p(y|x^*, \mathcal{D})]$ , where  $\mathbb{E}_{p(\omega|\mathcal{D})}p(y|x^*, \mathcal{D})$  denotes the average of predictions.

## 4.4 Experiments and Results

### 4.4.1 3D MRI Brain Tumor segmentation

*Data Description:* Our proposed framework is experimentally evaluated on the multimodal Brain Tumor Segmentation challenge data (BRATS2020 data set) published by the Center for Biomedical Image Computing and Analytics (CBICA), University of Pennsylvania [97–99]. It contains clinically acquired, multi-institutional pre-operative multimodal MRI scans with accompanying tumor sub-region delineations. For each patient, 4 MRI modalities, specifically, T1-weighted (T1), T1-weighted contrast-enhanced (T1ce), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes are provided. Accompanying manually annotated ground truth labels are approved by neuro-radiologists and comprise the necrotic and non-enhancing tumor core (NCR/NET — label 1), the peritumoral edema (ED — label 2), the GD-enhancing tumor (ET — label 4), and label 0 for everything else. The segmentation task involves partitioning the nested glioma sub-regions and produce segmentation labels as whole tumor (WT) (label 1, 2, 4), tumor core (TC) (label 1, 4) and enhancing tumor (ET)(label 4). Data is co-registered to a common anatomical template with isotropic resolution  $1mm^3$  and skull stripped. Images has the dimension  $240 \times 240 \times 155$  where 155 is the number of slices in the axial direction. Data set is consisted of MRI scans from 369 subjects in the training set and 125 subjects in the validation set.

*Data Pre-processing:* Data set consisting of 3D input images are too large to fit into GPU memory. All the input images were cropped into a size of  $128 \times 128 \times 128$  voxels.

*Model Architecture:* Segmentation performance of encoder-decoder like networks with various architectural modifications have been experimentally investigated by several studies in past literature. However, in the seminal work by Isensee et al. [100], the authors show that such variants do not provide an additional benefit and a well-trained U-Net [101] can yield competitive results and is more effective. Therefore, we set a U-Net as the baseline segmentation model. Patches of size  $128 \times 128 \times 128$  from each 4 modalities are fed into the network as input and thus the input matrix has the shape  $4 \times 128 \times 128 \times 128$ . While the down-sampling layers store information in the images at progressively lower resolutions, up-sampling layers reconstruct the segmentation map at the original resolution. The output has the shape  $3 \times 128 \times 128 \times 128$  where each  $128 \times 128 \times 128$  probability matrix represents the likelihood of each voxel belonging to one specific tumor sub-region. The proposed probabilistic approach is implemented on top of the baseline U-Net architecture, depicted in Figure 4.4.

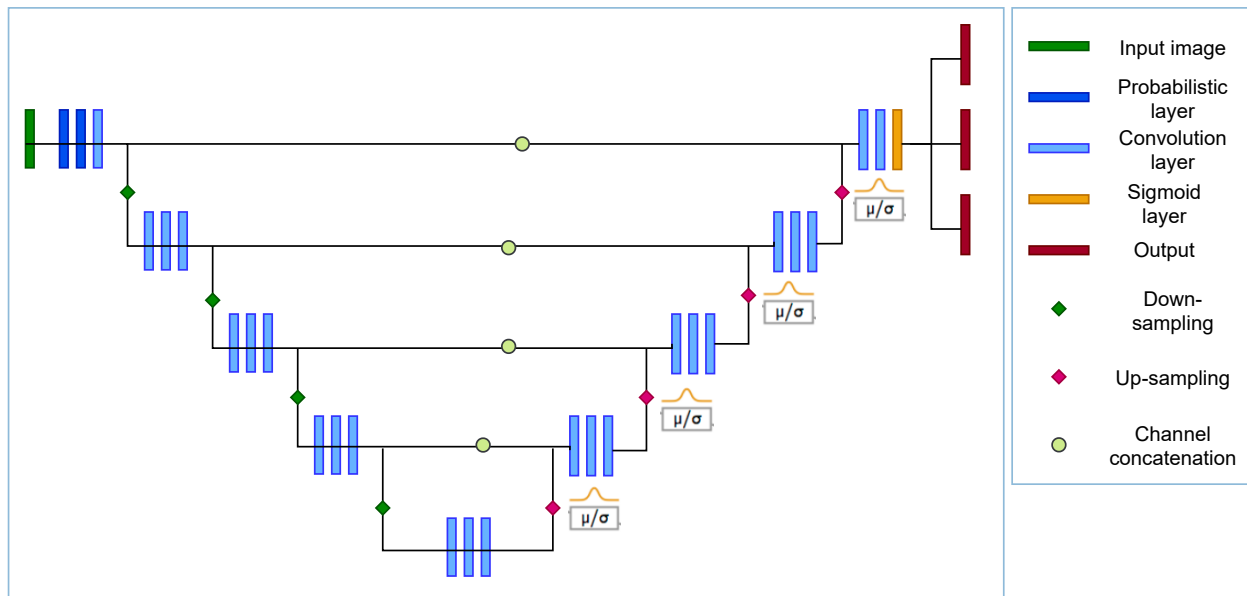


Figure 4.4.: Schematic U-Net like network architecture of the proposed probabilistic model.

While the provided expert annotated segmentation labels include 4 classes as (NCR/NET), ED, ET and background, the segmentation evaluation is performed on 3 nested sub-regions: WT, TC and ET. Prior work [100, 102–105] suggest that direct optimization for 3 sub-regions enhance performance. We therefore adopt a multi-task approach where per-voxel classification for each

tumor region as WT, TC and ET is considered a binary classification task, hence sigmoid activation  $\mathbb{R} \rightarrow [0, 1]$  is used in the last layer. Sigmoid over logits  $z$  is computed as  $\text{sigmoid}(z) = \frac{1}{1+\exp(-z)}$ .

*Model Training:* Using the combined loss defined in Eq. 4.9 and with an initial learning rate of  $10^{-4}$ , model was trained for 100 epochs. If the validation loss plateaued for 5 epochs then the learning rate was reduced by a factor of 0.5.

#### 4.4.2 Quantitative Evaluation

Both training and validation data set performance were evaluated through the online evaluation platform by CBICA, the Image Processing Portal (IPP).

*Segmentation measures:* Segmentation performance is evaluated subject to four metrics: Dice score, sensitivity, specificity and Hausdorff distance (95% percentile). The Dice similarity is a measure of pixel or voxel overlap between the ground truth and predicted regions. Sensitivity (recall or the true positive rate) measures the correct overlap of tumor regions, whereas specificity (actual negative rate) measures the correct overlap of non-tumor regions. Hausdorff distance evaluates the maximum distance between the segmentation and ground truth boundaries. These criteria is computed as follows:

$$\begin{aligned}
 \text{Dice}(P, T) &= \frac{2|P_1 \cap T_1|}{|P_1| + |T_1|} \\
 \text{Sensitivity}(P, T) &= \frac{|P_1 \cap T_1|}{(|T_1|)} \\
 \text{Specificity}(P, T) &= \frac{|P_0 \cap T_0|}{|T_0|}
 \end{aligned} \tag{4.12}$$

$$\text{Hausdorff}95(P, T) = \max\{\sup_{p \in P_1} \inf_{t \in T_1} d(p, t), \sup_{t \in T_1} \inf_{p \in P_1} d(t, p)\},$$

where  $P_1$  is the predicted tumor region,  $T_1$  is the actual tumor regions,  $P_0$  is the predicted non-tumor region and  $T_0$  is the actual non-tumor regions. In Hausdorff distance formula, sup represents the supremum and inf, the infimum.

*Uncertainty performance evaluation:* Uncertainty maps associated with the voxel labels are created for each tumor sub-region WT, TC and ET. Uncertainty values are normalized between 0 and 100 denoting the confidence of per-voxel classification from most confident (0) to most uncertain (100). Model performance can be assessed by filtering out uncertain voxels at different thresholds,  $T(0 \leq T \leq 100)$ , and calculating the Dice score on the remaining voxels. To monitor the true positives (TP) and true negatives(TN) that are filtered out at each  $T$ , the ratio of filtered TP

(FTP) and filtered TN (FTN) is calculated. FTP is calculated as  $FTP = (TP_{100} - TP_T)/TP_{100}$ , where  $TP_T$  represents true positives at threshold  $T$ . FTN is calculated similarly. Uncertainty is evaluated based on the area under the curve (AUC) with respect to: (1) Dice vs  $T$ , (2) FTP vs  $T$ , (3) FTN vs  $T$ , for different threshold values  $T$ , which are referred to as Dice AUC, filtered true positive (FTP) ratio AUC and filtered true negative (FTN) ratio AUC.

#### 4.4.3 Experimental Results

In this subsection, we report results of performance evaluations of the proposed method against the benchmark methods.

*Qualitative Results:* Figure 4.5 depicts the qualitative results for a sample of four subjects in BRATS2020 data-set, evaluated with deterministic, MC Dropout and probabilistic SegNet. While predictions are similar to ground truth for most pixels and show high confidence, tumor contours are associated with high uncertainty which is reasonable since the tumor boundaries are difficult to discern. Deviations of predicted segmentation’s from the ground truth, primarily in the central part of the tumor, are consistent with the regions of high uncertainty. It is worth to notice that in row 3, both deterministic and MC dropout model mis-classify the tumor region as tumor core. Only the probabilistic SegNet yields a closer prediction to that of ground truth and depict high uncertainty in the regions the network is uncertain about. The three models demonstrate notably different levels of uncertainty.

*Quantitative Results:* The quantitative results of segmentation performance and uncertainty estimation for the deterministic, MC Dropout and probabilistic models are reported in Table 4.1 and Table 4.2, respectively. Box plots for the segmentation performance on validation data are depicted in Figure 4.6. Uncertainty maps with the MC dropout method is generated through 10 stochastic forward passes and estimating voxel-wise standard deviation. In terms of both the segmentation and uncertainty performance, the proposed probabilistic deep network outperforms or in par with the other methods.

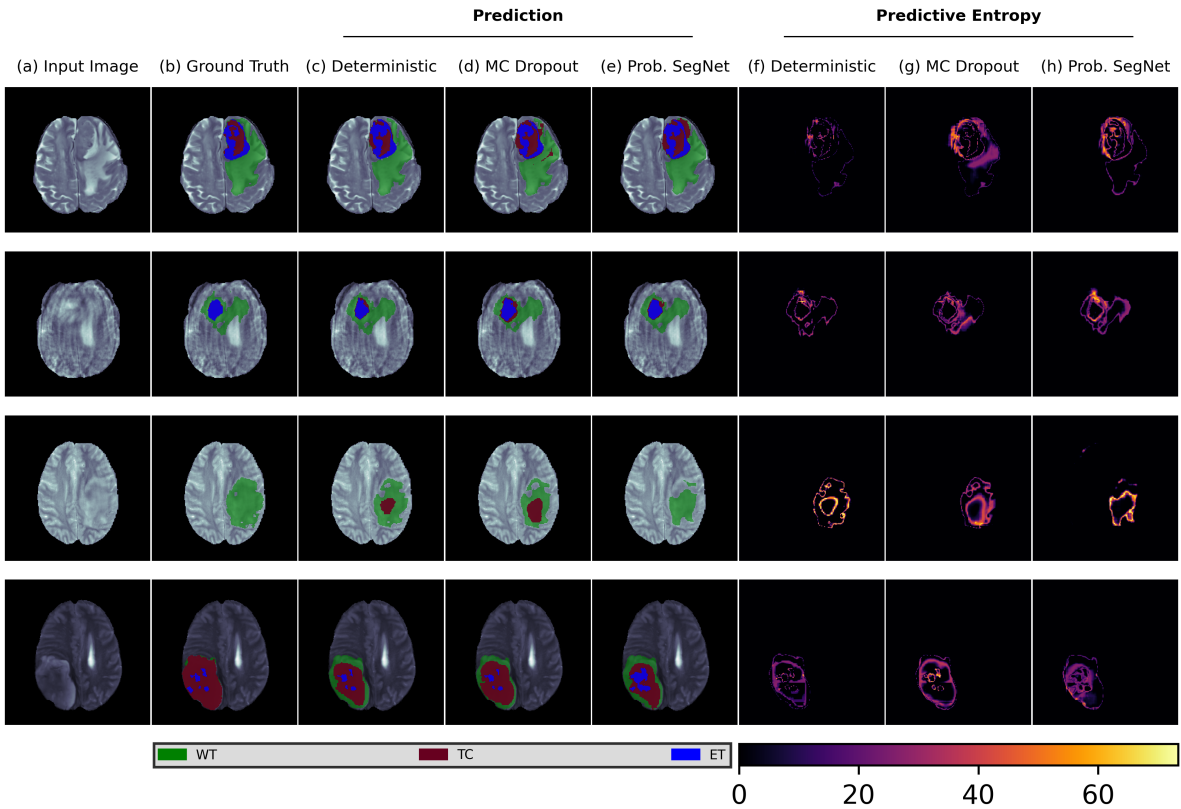


Figure 4.5.: Qualitative results of four representative subjects on the BRATS 2020 data set: (a) T2 slice; (b) Ground-truth segmentation; Prediction of (c) deterministic model, (d) MC Dropout model, and (e) proposed probabilistic SegNet; Predictive entropy of (f) deterministic model, (g) MC Dropout model, and (h) proposed probabilistic SegNet.

Table 4.1: Segmentation Performance Evaluation: Results on BRATS2020 training and validation sets as generated by the online portal

Training Set												
Method	Dice Score			Sensitivity			Specificity			Hausdorff95 Score		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
U-Net	0.7584	0.8945	0.8319	0.7756	0.9009	0.8413	0.9997	0.9991	0.9994	26.7559	7.5559	11.1538
MC-dropout	0.7608	0.8983	0.8271	0.7584	0.8993	0.8679	0.9997	0.9991	0.9992	24.4996	6.4273	10.1783
Probabilistic SegNet	0.7699	0.8987	0.8613	0.7840	0.9058	0.8710	0.9995	0.9996	0.9993	23.6278	6.9347	10.1489
Validation Set												
Method	Dice Score			Sensitivity			Specificity			Hausdorff95 Score		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
U-Net	0.6937	0.8791	0.7771	0.6976	0.9019	0.7743	0.9997	0.9989	0.9995	44.5283	8.8968	18.3084
MC-dropout	0.6982	0.8782	0.7756	0.6907	0.8833	0.7953	0.9997	0.9991	0.9993	43.2667	5.9141	16.7405
Probabilistic SegNet	0.6821	0.8718	0.7185	0.7130	0.9090	0.7273	0.9996	0.9992	0.9993	46.5851	16.5599	26.5780

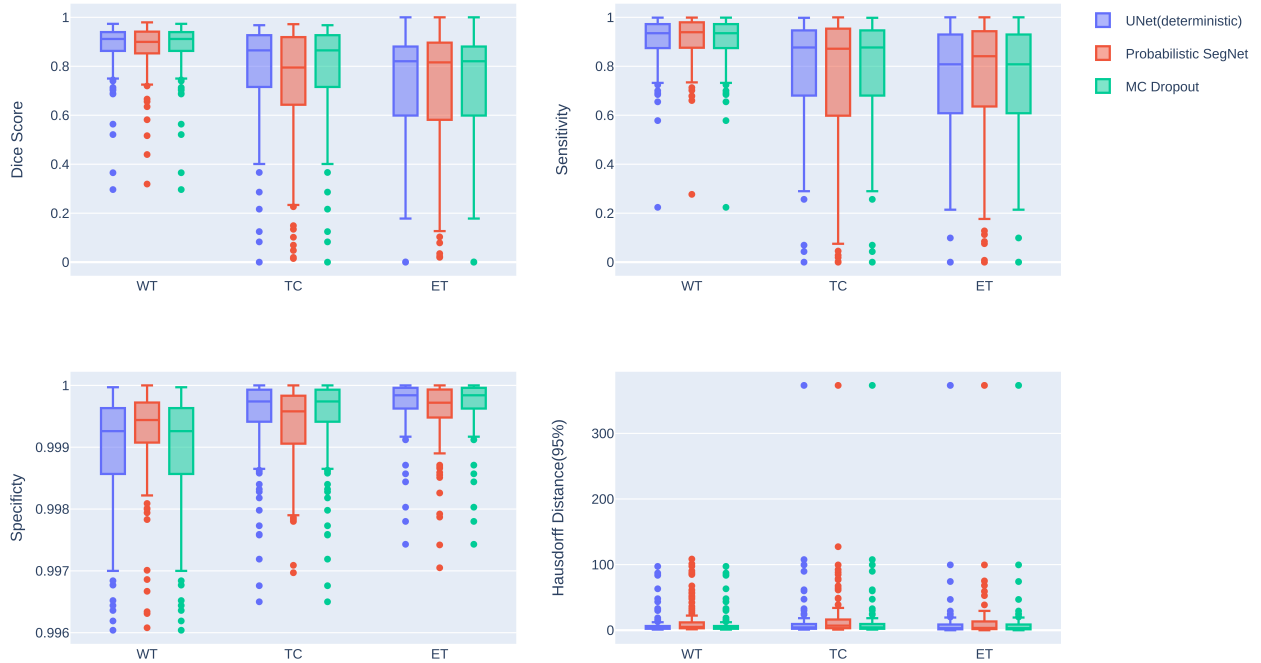


Figure 4.6.: Evaluation of the mean segmentation performance of the probabilistic SegNet and comparison against deterministic U-Net and MC dropout modeling approaches on validation data. Boxes extend values extending from first to third quartile. The central line in each box indicates the median and outliers are represented with dots.

Table 4.2: Uncertainty Estimation Performance Evaluation: Results on BRATS2020 training and validation sets as generated by the online portal

Training Set									
Method	Dice AUC			FTP Ratio AUC			FTN Ratio AUC		
	WT	TC	ET	WT	TC	ET	WT	TC	ET
MC-dropout	0.8867	0.8277	0.7613	0.0051	0.0051	0.0031	0.0001	0.0001	2.42E-05
Probabilistic SegNet	0.8984	0.8626	0.7711	0.0073	0.0070	0.0026	0.0003	0.0001	7.49E-05
Validation Set									
Method	Dice AUC			FTP Ratio AUC			FTN Ratio AUC		
	WT	TC	ET	WT	TC	ET	WT	TC	ET
MC-dropout	0.8784	0.7664	0.6928	0.0062	0.0052	0.0032	0.0001	0.0001	2.60E-05
Probabilistic SegNet	0.8825	0.7745	0.6988	0.0072	0.0072	0.0030	0.0002	0.0002	6.34E-05

#### 4.5 Contributions and Concluding Remarks:

- We presented an approach for predictive uncertainty estimation of bio-medical image segmentation utilizing Bayes by Backprop method under the variational inference framework.
- We evaluated and compared the quality of uncertainty gained using MC dropout at test time and frequentist inference to that achieved from probabilistic back-propagation.
- While the segmentation performance with the variational inference approach is in par with the deterministic network, it additionally incorporates the two desiderata imperative for safety-critical applications, a measure of uncertainty and regularization.
- Through empirical evaluations on complex neuro-imaging data we demonstrated that the probabilistic deep learning scheme naturally handles uncertainty and regularization.

**CHAPTER 5**  
**RADIOMICS IN NEURO-ONCOLOGY: A SPARSE BAYESIAN APPROACH**  
**FOR MODELING HIGH-DIMENSIONAL DATA IN GLIOBLASTOMA**  
**SURVIVAL PREDICTION**

**5.1 Introduction**

In epidemiological and clinical studies, the analysis of time until the occurrence of an event of interest such as the time from diagnosis of a disease until recurrence or death after some treatment is particularly common. Event-time analysis has applications in a range of disciplines including medicine, engineering, and econometrics. While standard regression procedures could be applicable for time-to-event modeling when censoring is not present, it may not be adequate as time to event is restricted to be non-negative and the distributions are often positively skewed. The main focus in this study is on event-time regression involving high-dimensional data where we analyze radiomic-based imaging data for survival prediction.

Data is said to be high-dimensional when the number of covariates  $p$  exceeds the number of observations  $n$ , often written as  $p \gg n$ . High dimensional data is common in many scientific disciplines such as genomics in computational biology, where identification of a subset of covariates associated with the response is particularly important. Unless properly handled, high dimensionality may lead to over-fitting and raise statistical issues in the analysis. While there are several well-established sparse regression approaches, Bayesian methods are favorable due to their ability to produce probability estimates over the model parameters enabling uncertainty quantification. This is even more important in clinical research where data acquisition is expensive, and the number of observations are limited. In the presence of limited data, model fits are subject to more variation resulting in high uncertainty.

While medical images are widely adopted in clinical research to assess biophysical properties of tumor cells, imaging-based studies often require analysis in the high-dimensional space. Radiomics, a method to extract clinically important features from high-dimensional clinical images, is com-



monly used to perform image-based tumor phenotyping. Radiomics research constitutes a relatively new area and its emerging role in neurology as a quantitative imaging biomarker has been influential. However, clinical imaging-based survival prediction is challenging, and high dimensionality poses additional challenges.

In this study we focus on predicting survival time and conducting risk factor analysis with pre-operative scans of patients with Glioblastoma Multiforme (GBM). Gliomas are a type of brain tumors that originate in glial cell with glioblastoma multiforme being the most aggressive, in which tumor cells infiltrate surrounding healthy brain tissues. Despite extensive studies to better understand the GBM cancer biology, owing to its genetic and clinical heterogeneity among patients, prognosis remains poor. Since it is clinically important to examine whether radiomic phenotyping has incremental prognostic value over the other clinical information, we aim to analyze the performance of GBM patient survival through radiomic feature-based statistical models combined with clinical features. Due to the natural handling of uncertainty through probabilistic interpretations of parameters, we adopt a sparse bayesian regression-based approach for survival modeling. In this study we present a sparse regression-based Bayesian AFT model in the context of radiomics survival prediction. Sparsity is induced across predictors using a shrinkage prior.

## 5.2 Related Work

*Radiomics:* Recent studies in bio-medical literature have reported performance improvements on survival prediction when radiomic features are incorporated. For example, findings in the study by Burgh et al. [106] suggest MRI features provide added value when predicting Amyotrophic lateral sclerosis (ALS) patient survival category as short, medium or long term survivors. Rayner et al. [107] demonstrated analyses on routinely collected CT images yield promising results when predicting patient longevity. A study conducted by Bae et al. [108] explore the applicability of MRI radiomic features combined with clinical and genetic profiles to predict the survival of GBM patients. They extracted 796 radiomic features from MRI scans, out of which 18 features were identified as significant. Nie et al. [109] adopted a 3D CNN to extract features from pre-operative brain images and in conjunction with clinical features they predicted survival of glioma patients via a support vector machine. However, features extracted through deep learning are abstract and the predictors are unknown, hence the degree of bias in the predictors cannot be determined. Prior research on glioma also suggest that patient age alone can predict the survival of the patient relatively well when compared with methods that incorporate more complex radiomic features [110],

[111]. While this can be considered a first step towards a more profound understanding of survival factors, additional studies are required to understand the key tenets of GBM survival prediction more completely.

*Sparse Learning:* Among the sparse learning methods for addressing high-dimensionality, regularization of the objective function through a penalized term is a common approach. For instance, least absolute shrinkage and selection operator (lasso) [112] impose L1-norm penalty on the regression parameters which shrinks them towards zero. Conversely, Bayesian penalization encode sparsity through the prior distribution. In high-dimensional survival data, the Laplace prior is commonly used [113–115], which is the Lasso equivalent under the Bayesian framework. The gold standard in sparsity priors is the spike-and-slab-prior [116] which is a discrete-continuous mixture, more specifically a point mass at zero (spike) and a diffused distribution (slab), but it is computationally inefficient in practice. A more recent prior, referred to as the horseshoe prior [117] received considerable attention within the research community, however it’s configuration is complicated. Heavily influenced by the work in [118], we incorporate a continuous relaxation of the spike-and-slab prior in our Bayesian AFT model, which is described in more detail in Section 4 .

### 5.3 Background, concepts and notation

#### 5.3.1 Survival analysis preliminaries

Let survival data of size  $n$  be denoted by  $\mathcal{D} = \{\mathbf{x}_i, t_i, \delta_i\}_{i=1}^n$ , where  $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}] \in \mathbb{R}^p$  are the covariates and  $(t_i, \delta_i)$  are the event pairs such that  $t_i$  indicates the time-to-event of interest and  $\delta_i$  is the binary event indicator. Typically,  $\delta_i = 0$  for a censored instance and  $\delta_i = 1$  for an event that is observed. The survival function  $S(t)$  indicates the probability that the event of interest does not occur within the observation window that ends at time  $t$  and can be characterized by,

$$S(t) = P(T \geq t) = 1 - F(t) = \exp\left(-\int_0^t h(s)d(s)\right), \tag{5.1}$$

where  $T$  is the survival time which is non-negative and continuous. For censored instances  $T$  is latent.  $F(t)$  is the cumulative distribution function of the event of interest which represents the probability that the event occurs within  $t$  days.

Alternatively, the probabilistic behavior of survival time  $T$  can be characterized by its hazard function and density function. The conditional hazard rate function  $h(t|\mathbf{x})$  represents the instan-

taneous rate of the event occurrence at time  $t$  given covariates  $\mathbf{x}$ , for the population group that is still at risk at time  $t$  which is defined as:

$$h(t|\mathbf{x}) = \lim_{dt \rightarrow \infty} \frac{P(t \leq T \leq t + dt)}{P(T \geq t|\mathbf{x})} = \frac{f(t|\mathbf{x})}{S(t|\mathbf{x})}, \quad (5.2)$$

where  $f(t|\mathbf{x})$  is the conditional density of the survival function and  $S(t|\mathbf{x})$  is the complement of the cumulative conditional density,  $F(t|\mathbf{x})$ .

### 5.3.2 Time to Event Regression: Accelerated Failure Time (AFT) model

Among the most common modeling frameworks to characterize the association between the event time and the covariates are the cox proportional hazards (CPH) model [119] and the accelerated failure time (AFT) [120]. Contrary to CPH which models the conditional hazard function  $h(t|x)$  through a semi-parametric approach, AFT adopts a parametric approach to model the survival time. Since the focus of current study is on survival time modeling, we briefly describe here the preliminaries pertaining to the AFT model.

The AFT model assumes that the explanatory variables have a multiplicative effect with respect to the survival time of the individual where the survival function is expressed as:

$$S(t|\boldsymbol{\beta}, \mathbf{x}) = S_0(\exp(\mathbf{x}^T \boldsymbol{\beta}) t), \quad (5.3)$$

where  $S_0(t)$  is the baseline survival function and  $\mathbf{x}$  are the covariates. If  $\mathbf{x}^T \boldsymbol{\beta} > 0$ , then  $\exp(\mathbf{x}^T \boldsymbol{\beta}) t > t$  and the covariates have an accelerative effect on the effective passage of time for the subject. In this case,  $\exp(\mathbf{x}^T \boldsymbol{\beta}) t$  is termed the accelerator factor.

Under the AFT model, survival time  $T$  is expressed in logarithmic scale and characterized by a log-linear regression model:

$$\log(T_i) = \mathbf{x}_i^T \boldsymbol{\beta} + \sigma \varepsilon_i, \quad i = 1, \dots, n \quad (5.4)$$

where  $\log(T_i)$  is the logarithm of survival time,  $\mathbf{x}$  is the  $p$ -vector of covariates,  $\boldsymbol{\beta} \in \mathbb{R}^p$  is the vector of regression coefficients indicating the degree of influence of covariates on the response,  $\sigma$  is a scale parameter and  $\varepsilon$  is the independently and identically distributed (iid) random error, assumed to follow a particular distribution.

The distribution of time-to-event  $T$  is determined by the distribution of error term  $\varepsilon$ . For instance, if the log-linear error distribution is normal, then  $T$  follows a log-normal distribution, whereas a logistic distribution for  $\varepsilon$  corresponds to a log-logistic distribution for  $T$ .

## 5.4 Sparse Bayesian Survival Regression

### 5.4.1 Bayesian AFT model formulation

While there exists a considerable body of literature on parametric time-to-event regression, research on AFT models under the Bayesian paradigm remains limited. Considering AFT models under Weibull distributional assumptions here we present a sparse Bayesian framework for the case  $p \gg n$ , where  $p$  represents the number of covariates and  $n$ , the number of observations.

Consider the following AFT model:

$$Y = \log(T) = \mathbf{x}^T \boldsymbol{\beta} + \sigma \varepsilon. \quad (5.5)$$

Suppose the  $\varepsilon$  in AFT model defined in (5.5) follows an extreme value distribution, more specifically a Gumbel distribution,  $G(0, 1)$ . Then  $T$  follows a Weibull distribution,  $W(\alpha, \gamma)$  where  $\alpha$  is the scale parameter and  $\gamma$  is the shape parameter.

Gumbel distribution has probability density function (pdf) of the form:

$$f_\varepsilon(u) = \exp(-u) \exp[-\exp(u)] \quad (5.6)$$

And cumulative distribution function(cdf):

$$F_\varepsilon(u) = 1 - \exp[-\exp(u)]. \quad (5.7)$$

Likelihood of the observed survival times can then be formulated as follows:

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma; \mathbf{y}_i) &= \prod_{i=1}^n [f_Y(y_i)]^{\delta_i} [S_Y(y_i)]^{1-\delta_i} \\ &= \prod_{i=1}^n \left\{ \frac{1}{\sigma} \exp\left(-\frac{y_i - \mathbf{x}_i^T \boldsymbol{\beta}}{\sigma}\right) \exp\left[-\exp\left(-\frac{y_i - \mathbf{x}_i^T \boldsymbol{\beta}}{\sigma}\right)\right] \right\} \left\{ \exp\left[-\exp\left(-\frac{y_i - \mathbf{x}_i^T \boldsymbol{\beta}}{\sigma}\right)\right] \right\}, \end{aligned} \quad (5.8)$$

where  $f$  and  $S$  represent density and survival function for the error distribution, respectively. The censoring indicator for subject  $i$  is represented by,  $\delta_i$ .

### 5.4.2 Prior Specification

A priori expectations of model parameter behavior can be encoded via a prior distribution on  $\beta$ . The aim of establishing a subset of coefficients to be exactly zero a priori may be realized through a shrinkage prior. Marginal prior distribution for regression coefficients,  $p(\beta)$  is specified by imposing the sparsity assumption to provide shrinkage to the small effects while allowing strong effects to remain large under the Bayesian posterior. Inspired by the previous work in literature on Bayesian lasso regression [121, 122], we define the prior based on the Laplace (double exponential) distribution for which the conditional density has the following form:

$$p(\beta_j|b^2) = \frac{1}{2b} \exp\left(\frac{-|\beta_j|}{b}\right), \quad (5.9)$$

where  $b$  is a scale parameter.

We define our shrinkage prior as a Laplacian scale mixture. The hierarchical representation of our shrinkage prior can be summarized as follows:

$$\begin{aligned} \beta_j|\tau, \lambda_i &\stackrel{iid}{\sim} LP(0, (\lambda_i\tau)^2), \quad j = 1, \dots, p \\ \lambda_i &\sim \text{LogitNormal}(\mu_\lambda, \sigma_\lambda) \\ \mu_\lambda &= \text{logit}(\theta) \\ \tau &\sim \text{HalfNormal}(\sigma), \end{aligned} \quad (5.10)$$

where  $\theta$  is a probability representing our prior beliefs of a variable being non-zero and  $\lambda = \text{logit}^{-1}(\tilde{\lambda}_i)$  such that  $\tilde{\lambda}_i \sim N(\mu_\lambda, \sigma_\lambda^2)$ .

The scale mixture parameter  $\lambda_i$  acts as an indicator of the inclusion of a variable. Thus, the values of the logit-normal random variable can be considered as approximation for the variable inclusion probabilities.

### 5.4.3 Model fitting and Posterior Inference

Bayes Theorem combines prior probability distribution with the likelihood that is derived based on the observed data. Inference concerning model parameters can be performed through the posterior distribution.

$$P(\beta, \sigma; y_i) = \frac{L(\beta, \sigma; y_i) p(\beta)}{\int_{\beta} L(\beta, \sigma; y_i) p(\beta) d\beta}, \quad (5.11)$$

where  $m(\mathcal{D}) = \int_{\beta} L(\beta, \sigma; y_i) p(\beta) d\beta$  is the model evidence which is analytically intractable and thus require sampling-based approaches such as MCMC to sample from  $p(\beta|\mathcal{D})$  at the inference phase.

## 5.5 Application to survival prediction of patients with Glioblastoma

### 5.5.1 Experimental Set-up

*Description of Data:* The proposed approach is experimentally evaluated on the multi-modal Brain Tumor Segmentation Challenge data (BRATS2020) published by the Center for Biomedical Image Computing and Analytics (CBICA) from University of Pennsylvania [97–99]. The data set is consisted of clinically acquired multi-modal pre-operative MRI scans of patients diagnosed with glioma tumors. The 3D MRI scans include 4 modalities; T1-weighted, contrast-enhanced T1-weighted (T1ce), T2-wieghted and T2 Fluid Attenuated Inversion Recovery (FLAIR) sequences. Manually segmented tumor sub-region labels by expert radiologists, age, resection status and survival information are provided for a cohort of 235 patients in the training set (Table 5.1). The data-set does not contain censored observations. Out of the patients who underwent surgical intervention, 118 had gross total resection (GTR) and 10 had sub-total resection (STR). The extent of resection status is reported as unknown (NA) for 107 patients.

Table 5.1: Statistical information for the training set in BRATS2020 survival data.

Variables	Value
Resection Status (cases)	
Gross Total Resection (GTR)	118
Sub-total Rescetion (STR)	10
Missing Information	107
Age (years)	
Range	18-87
Mean $\pm$ std	61 $\pm$ 11.9
Median	61
Overall survival time (GTR cases)	
Short-term ( <10 months)	42
Medium-term (between 10 and 15 months)	30
Long-term ( >15 months)	46

*Performance Evaluation:* Survival predictive performance is evaluated using mean square error (MSE). Results for median squared error (median SE) and standard deviation of the squared error (std SE) will also be presented which can be obtained by computing the median and standard deviation of the estimator's squared errors, respectively. Mean squared error of the predicted survival times is defined by:

$$MSE = \frac{1}{N} \sum_{i=1}^N (t_i - \hat{t}_i)^2, \quad (5.12)$$

where  $t_i (i = 1, \dots, N)$  is the actual survival times and  $\hat{t}_i (i = 1, \dots, N)$  is the predicted times.

### 5.5.2 Exploratory Analysis

#### (a) Kaplan-Meir Survival Estimates

An initial exploratory analysis on the overall survival showed the median OS for the entire cohort of study is 370 days (approximately 12 months). Neurosurgical options for glioblastoma include removal of complete tumor area which is referred to as Gross Total Resection (GTR) or removal of part of the tumor, referred to as Sub-total Resection (STR). We first seek to examine the association between the extent of tumor resection and survival time. The median duration of overall survival in the GTR and STR groups are 375 days and 652 days, respectively. Sub-group analysis by resection-status was performed through the Kaplan-Meir (KM) estimates (Figure 5.1) where the survival function is represented by:

$$\hat{S}(t) = \prod_{j:T_j < t} \left(1 - \frac{d_j}{r_j}\right), \quad (5.13)$$

where  $T_j$  represents distinct event times,  $d_j$  is the number of events that occurred at  $T_j$ ,  $r_j$  is the number of subjects "at risk".

In Figure 5.1, the two survival curves appear to overlap, but it can be seen that the probability of survival is low for the patients who underwent GTR than those who were treated with STR. However, patients with GTR status are associated with longer survival times and clearly have a better chance of surviving more than three years.

#### (b) Parametric characterization of survival time

We made comparisons among number of parametric models (over 30) and a model which appropriately characterize data was selected based on the the Akaike Information Criterion (AIC).

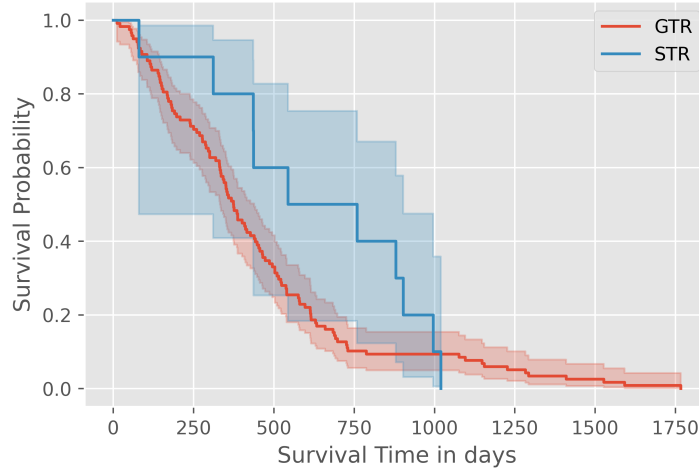


Figure 5.1.: Kaplan-Meier plots of overall survival in all cases by resection status.

AIC selects the relative quality of parametric model based on the maximum log likelihood and the principle of parsimony. Mathematically,  $AIC = -2LL + 2k$  where  $k$  is the number of model parameters and  $LL$  is the maximum log-likelihood. Taking the AIC score into consideration we chose the Gamma distribution, denoted by  $\Gamma(\alpha, \beta)$ , as an appropriate distribution to characterize overall survival times of patients with Glioblastoma. Goodness-of-fit was evaluated through the Kolmogorov-Smirnov test which yielded a p-value of 0.3549 indicating a good fit. The gamma distribution is parameterized by the shape parameter  $\alpha$  and an inverse of a scale parameter ( $\theta$ ), denoted by  $\beta = 1/\theta$ , which is often referred to as the rate parameter. The maximum likelihood estimates for the shape parameter ( $\alpha$ ) is 1.53 and scale parameter ( $\theta = 1/\beta$ ) is 289.89. Density and survival plots are displayed in (Figure 5.2). Survival function is non-increasing and allows to obtain the probability that a patient will survive past time  $t$ .

Let  $T$  be a random variable denoting the the survival time which is approximated by a Gamma distribution. Then for  $y \in \mathbb{R}^+$ , the fitted distribution has the following form:

$$Gamma(t|\alpha, \beta) = \frac{\beta(\beta t)^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}, \quad (5.14)$$

where  $\alpha \in \mathbb{R}^+$  and  $\beta \in \mathbb{R}^+$ .

Thus, the pdf of survival times for patients with Glioblastoma can be represented by,

$$f(t) = \frac{0.0034(0.0034t)^{1.53-1} e^{-0.0034t}}{\Gamma(1.53)}, \quad (5.15)$$



where  $\alpha$  is the shape parameter. The scale parameter is denoted by  $1/\beta$  which indicates the mean time between events.

Similarly, the survival function can be characterized by:

$$S(t) = 1 - I_\alpha(\beta t) \tag{5.16}$$

$$S(t) = 1 - I_{1.53}(0.0034t),$$

where  $I_\alpha$  is called the incomplete gamma function with the following form:

$$I_\alpha(t) = \int_0^t \beta^{\alpha-1} e^{-t} dt / \Gamma(\alpha). \tag{5.17}$$

The survival function does not have a closed-form expression. The hazard function can be obtained by computing  $f(t)/S(t)$ . Since the shape parameter  $\alpha = 1.53 > 1$ , the gamma hazard increases monotonically starting from 0 to a maximum value of  $\beta$ .

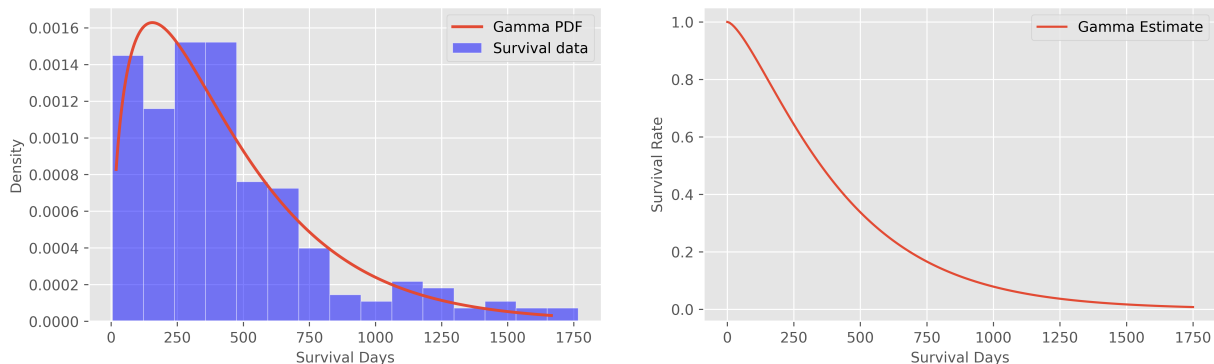


Figure 5.2.: Fitted (a) probability density and (b) survival functions with Gamma distribution for the entire cohort of patients with Glioblastoma BRATS2020 data set. Survival function represents the likelihood of survival after  $t$  days of GTR.

### 5.5.3 Radiomics analysis for survival time prediction

In the BRATS2020 challenge data, the ground truth values for validation set is not provided. Model performance metrics can be obtained by submitting the results to the online portal provided by CBICA Image Processing Portal [123] where the results are evaluated only on patients whose resection status is GTR. Therefore, we set out to study overall survival of patients with resection status GTR. The goal is to predict the number of survival days of patients on a validation cohort of 29 after Gross Total Resection (GTR) of malignant tumor.

*Process Overview:* Survival Prediction in clinical neuroimaging involves identification of tumor margins to estimate patient survival statistics, therefore the first step in radiomics is tumor segmentation. While segmentation can be performed via a variety of methods ranging from manual labeling to deep-learning methods, in this study we utilize the ground truth annotations provided in the data set for training, and for validation we used the segmentation maps we obtained using our own model. Next, radiomics features were extracted from clinical images which serve as quantitative imaging bio-markers. Then feature selection methods were applied to select the most relevant set of features. It is followed by rigorous statistical analysis to predict the survival of GBM patients. A graphical illustration of the radiomics workflow is provided in Figure 5.3. Details of each step of the radiomics workflow is summarized below.

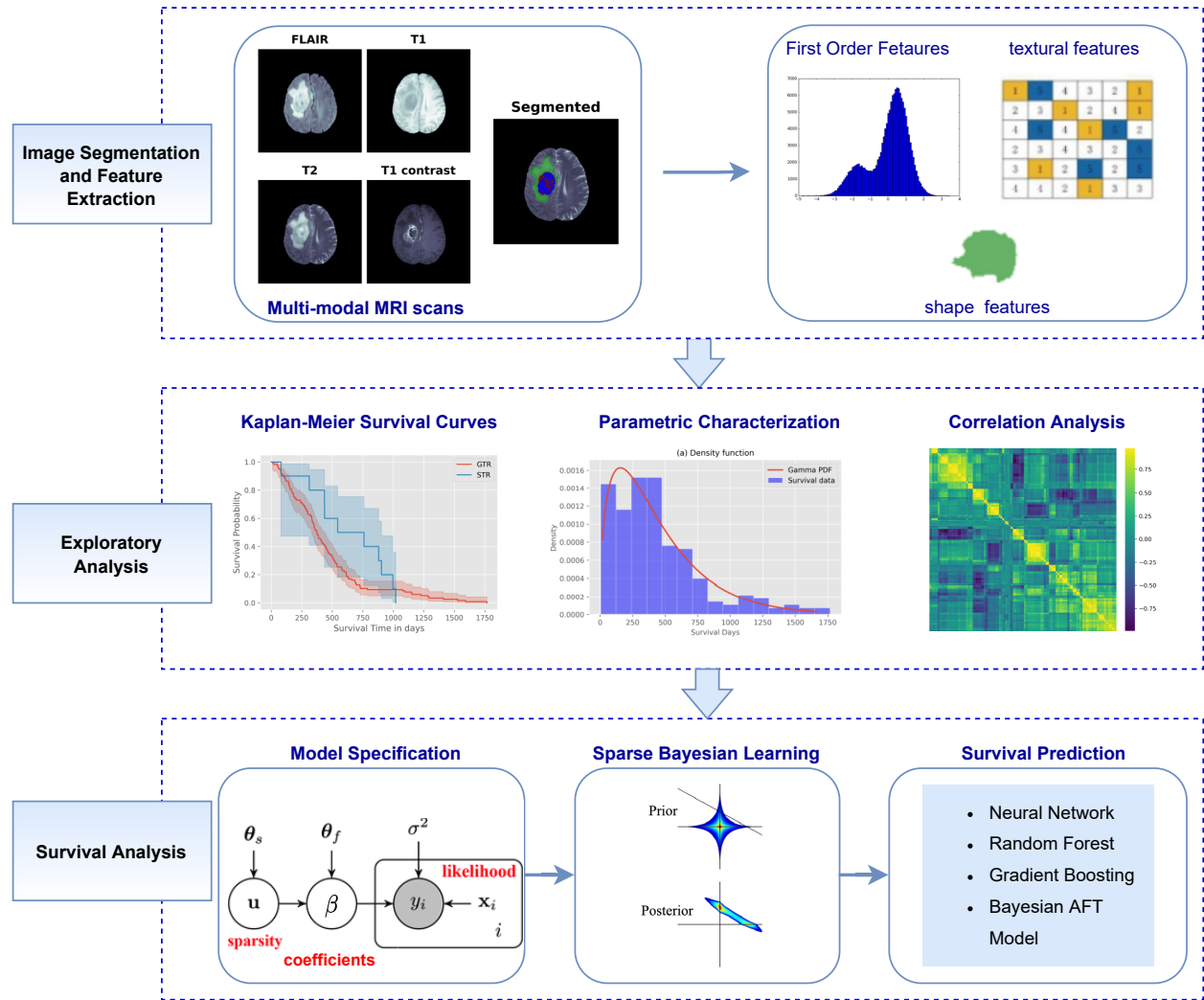


Figure 5.3.: Overview of radiomics workflow for Survival Prediction. Ground truth segmentation masks are used to extract radiomics features from the tumor regions. Extracted features are based on intensity, textural and shape. Then an exploratory analysis is conducted followed by the statistical analysis for survival prediction.

### 5.5.3.1 Quantitative Feature Extraction

Having performed discrete wavelet transform (DWT) on each MRI modality (T1, T1CE, T2 and FLAIR), the 4 modalities were fused using fusion rules (by taking the average). The process is displayed in Figure 5.4. Then using the fused image and the 3 regions of interest in the segmented scan (whole tumor, tumor core and enhanced tumor), the radiomics features were extracted using Pyradiomics [124] which is an open-source python package for radiomics feature extraction. Description of extracted features are included in Appendix A.

Extracted radiomics features include 2D and 3D shape-based features (10 features and 16 features, respectively), first order statistics (19 features), gray level co-occurrence matrix features (24 features), gray level run length matrix (16 features), gray level size zone matrix (16 features), neighbouring gray tone difference matrix (5 features), gray Level dependence matrix (14 features).

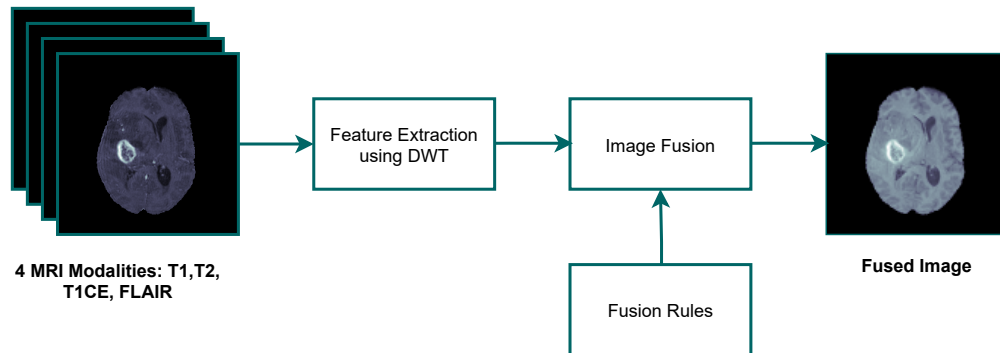


Figure 5.4.: An example of a fused image of 4 modalities in Glioblastoma BRATS2020 data set.

### 5.5.3.2 Statistical Analysis and model building

#### (a) AFT Model Selection:

Feature correlations are depicted in the cluster map in Figure 5.5. It is clearly visible that most variables have high-correlations among them. In order to select a suitable probability distribution for the AFT model, we first filtered the highly correlated variables, with threshold set to 0.95 where the number of features were reduced to 91 from 331. Note that for the Bayesian AFT model and the other methods we made comparisons with, we considered the full feature space.

Widely adopted probability distributions under the parametric AFT model are the Log-Normal, Log-logistic and Weibull distributions. Through some transformation of survival function a straight-line plot against log time can be generated which allows us to determine a suitable distribution for survival times. Survival function transformations are summarized in Table 5.2. In order to identify

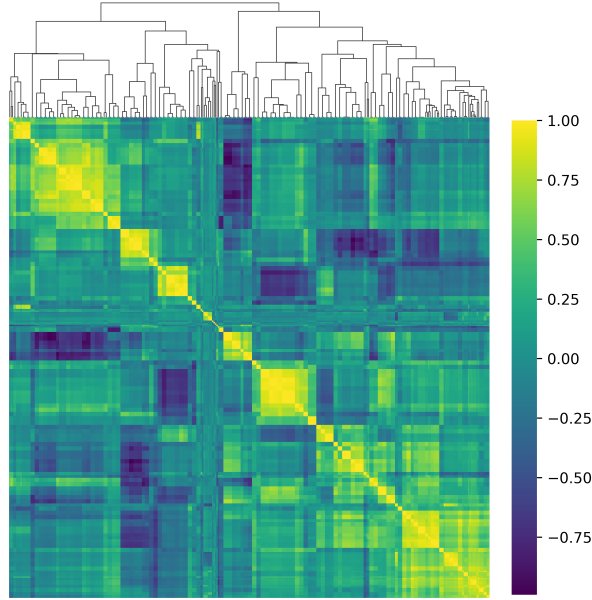


Figure 5.5.: The cluster-map map provides a clear visual representation of feature correlations. The dendrogram displays the clustering among features based on correlation. The color scale ranging from -1 to +1 indicates correlation coefficient.

the underlying distribution that best characterize the survival model, transformed survival functions were plotted against the log of survival time (Figure 5.6). Visual inspection of the transformed empirical survival curves indicates that the Weibull distribution is a good fit compared with the other considered distributions. This was further confirmed by the AIC score for which Weibull distribution had the lowest value. Therefore, we continue the analysis with the Weibull distribution.

Table 5.2: Survival Function Transformations

Distribution	Survival Function	Linear Transformation
Log-normal $(\mu, \sigma)$	$S(t) = 1 - \phi\left(\frac{\ln(t) - \mu}{\sigma}\right)$	$\phi^{-1}[1 - S(t)] = \frac{1}{\sigma}\ln(t) - \frac{\mu}{\sigma}$
Log-logistic $(\theta, k)$	$S(t) = \{1 + \exp(\theta)t^k\}^{-1}$	$\ln\frac{S(t)}{1-S(t)} = -\theta - k \ln(t)$
Weibull $(\alpha, k)$	$S(t) = \exp\{-(\alpha t)^k\}$	$\ln\{-\ln(S(t))\} = k\ln(\alpha) + k\ln(t)$

*(b) Model Training:*

Patient age is the only non-imaging clinical feature in the data set. Linear regression on the age of the patient has proven to be effective in prior literature. We therefore set a linear regression model for the log-transformed survival days with patient age as the only predictor as the baseline model. Additionally, we trained Random Forests and Neural Networks to compare the performance

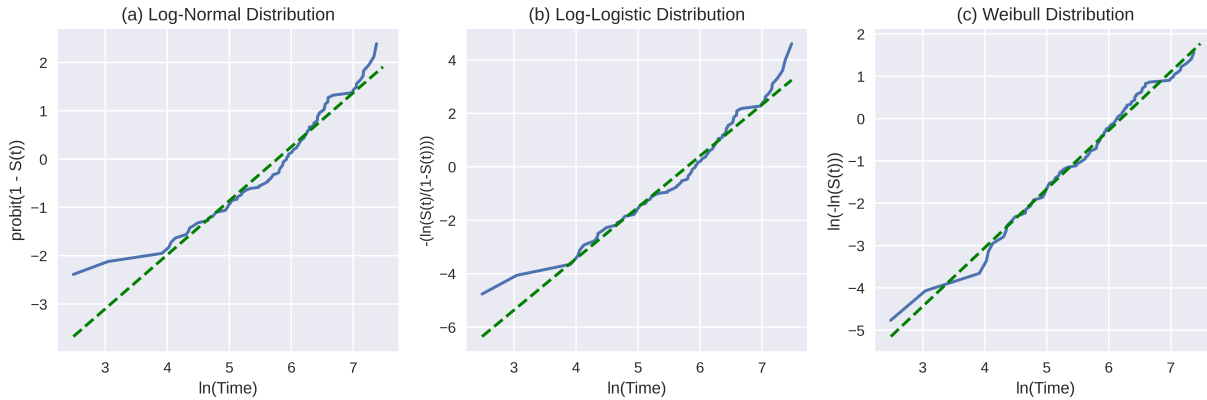


Figure 5.6.: Plots of transformed survival functions for (a) Log-Normal distribution, (b) Log-logistic Distribution and (c) Weibull distribution. Weibull distribution indicates a better fit compared with other distributions.

of the Bayesian AFT model. Random Forest regressor was trained with number of trees set to 1000 and mean squared error as split criterion. Two-layer Neural Network is trained for 400 epochs.

The importance of the Bayesian approach is evident in the Figure 5.7 where a Bayesian linear regression model with only age of the patient age as the predictor is fitted. We can see a range of regression lines as sampled by the Bayesian model. Under limited data, uncertainty is high, which is represented by the variability of the regression lines, implying the importance of uncertainty quantification.

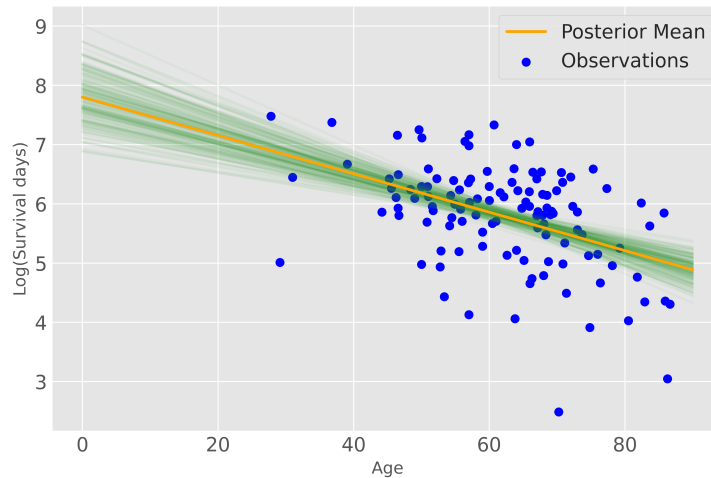


Figure 5.7.: Bayesian linear regression model with only age of the patient as a predictor. While OLS yields a single estimate of the model parameters, with Bayesian regression we can obtain a sample of credible regression lines which are demonstrated in green. In the regions with less data there is high uncertainty.

*Graphical Model:* The graphical model indicating the hierarchical representation of the Bayesian AFT model is given in Figure 5.8. It demonstrates the inter-dependence between the variables.

The nodes represent the variables, and the edges indicate the conditional dependence given other variables.

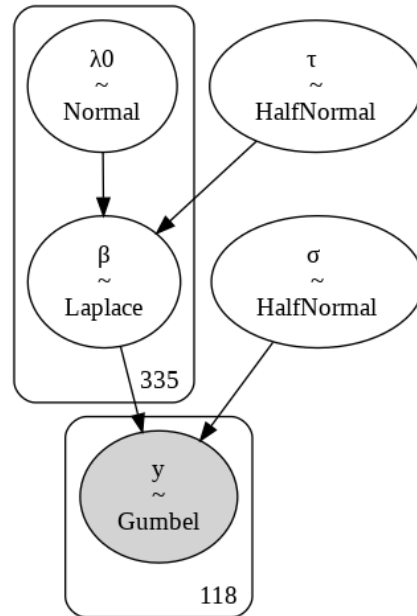


Figure 5.8.: Graphical model depicting the conditional dependencies.

#### 5.5.4 Experimental Results

*Convergence diagnostics:* Exact Bayesian Inference is not analytically tractable hence approximate inference based on Markov chain Monte Carlo (MCMC) is applied. We used the No-U-Turn Sampler(NUTS) [125] MCMC algorithm, which is an extension of the Hamiltonian Monte Carlo (HMC). Provided below in Figure 5.9 are the trace plots comprising of a set of sampled parameters over all chains conditioned on the observed data and priors. And Gelman’s Rubin score is close to 1 confirming convergence.

*Bayesian Variable Selection:* Identifying the relevant features is a key aim of the analysis. Variable selection can be attained via the marginal posterior inclusion probabilities. In Table 5.3, we present summary statistics of five potential factors identified through inclusion probabilities.

In Table 5.3, variable inclusion probability, mean, standard deviation(sd) and the Highest Density Interval (HDI) is provided. Among the most influential variables are patient age, proportion of tumor core (TC) to whole tumor (WT), enhancing tumor (ET) length and Gray Level Size Zone (GLSZM) features. GLSZM quantifies gray level zones in the MRI scan. High Density Interval(HDI) incorporates the most credible values and can be treated as the posterior distribution’s

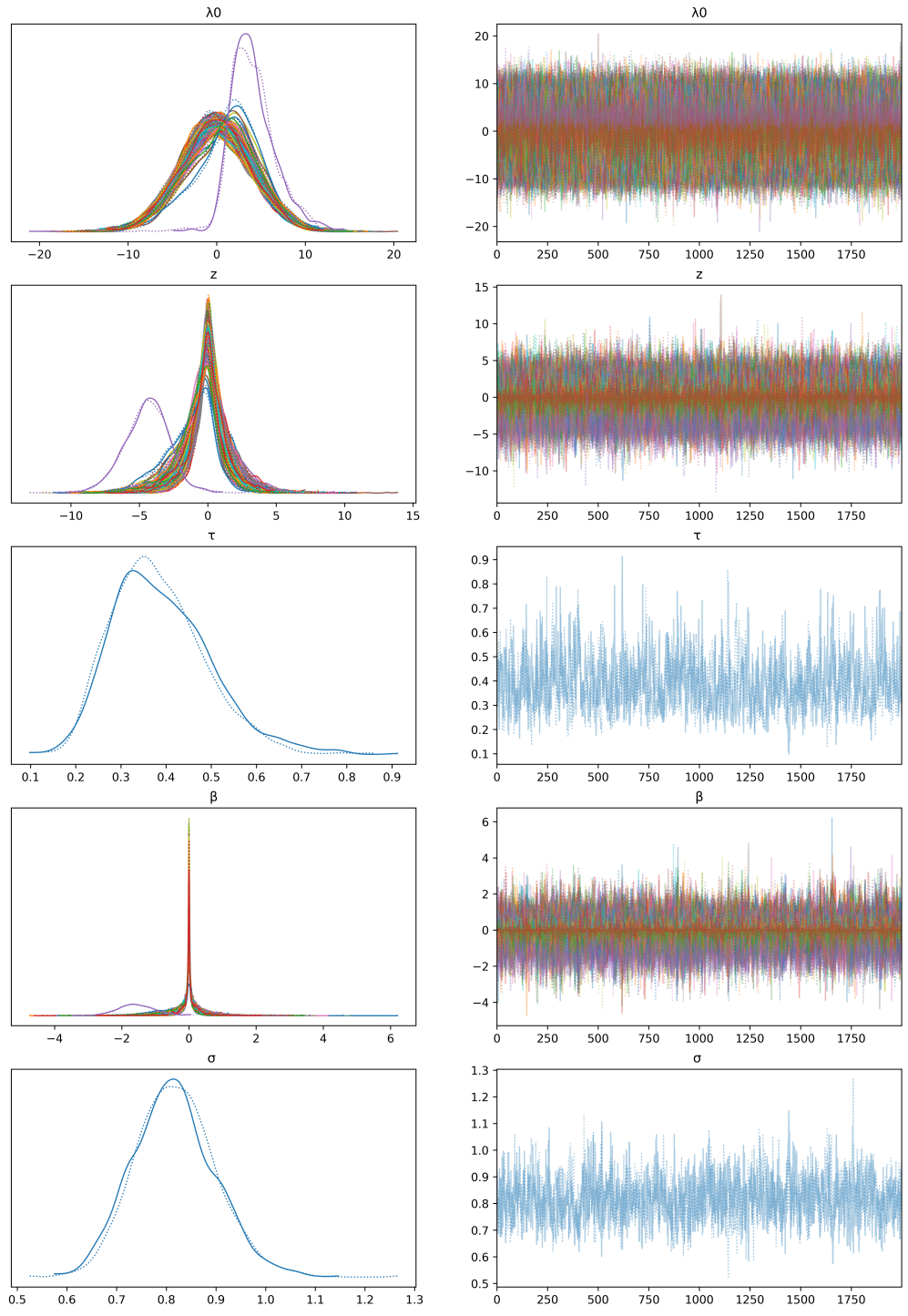


Figure 5.9.: Trace plots for parameters indicating convergence.

summary credible interval. Values within HDI has a greater probability density than the those that are outside the interval.

Table 5.3: Bayesian variable selection and their summary statistics. Variable inclusion probability, mean, standard deviation(sd) and their Highest Density Interval (HDI) is provided.

Variable	Inclusion probability	mean	sd	hdi_3%	hdi_97%
Age	0.9298	-1.615	0.533	-2.645	-0.637
Proportion_TC_WT	0.6431	-0.444	0.600	-1.772	0.247
Original GLSZM Gray Level Variance WT	0.5841	-0.341	0.597	-1.645	0.427
Original GLSZM Gray Level Variance ET	0.5818	-0.342	0.602	-1.664	0.411
Length_ET	0.5723	-0.310	0.549	-1.477	0.428
Original GLSZM Gray Level Variance TC	0.5679	-0.299	0.547	-1.474	0.349

Table 5.4: Glioblastoma survival prediction performance evaluation on BRATS2020 data set

Training Set			
	MSE	Median SE	Std SE
Baseline (Linear Regression on age only)	9.55e4	2.54e4	1.82e5
Two-layer Neural Network	9.31e4	2.83e4	1.76e5
Random Forest	1.66e4	5.75e3	3.13e4
Gradient Boosting	6.15e2	2.49e2	9.87e2
Bayesian AFT Model	2.88e4	1.57e4	5.42e4
Validation Set			
	MSE	Median SE	Std SE
Baseline (Linear Regression on age only)	8.90e4	3.31e4	1.21e5
Two-layer Neural Network	9.74e4	4.24e4	1.28e5
Random Forest	8.28e4	4.03e4	1.08e5
Gradient Boosting	8.62e4	2.74e4	1.13e5
Bayesian AFT Model	7.97e4	5.30e4	1.19e5

In Table 5.4, we have provided a comparison of survival prediction performance of Bayesian AFT model and other methods. It can be seen that while there is a considerable improvement on the predictive performance over the baseline, due to the absence of other relevant clinical information the predictive ability of the survival prediction performance is not high. Thus, the results suggest that radiomic features are potent in predicting survival time, but both imaging and other relevant clinical features need to be incorporated to gain a higher performance. While performance of the AFT model is comparable to the other machine learning methods, we emphasize that advantage of the Bayesian approach is not improving accuracy, but its ability to yield model parameter estimates concurrently with variable selection and estimations of uncertainty.



## 5.6 Contributions and Concluding Remarks

We presented a Bayesian AFT variable selection approach to address high-dimensional problem when modeling time-to-event data where the sparsity is induced by a hierarchical prior derived by exploiting the Bayesian Lasso shrinkage prior. Mixture models are particularly useful when developing sparsity inducing priors. Effective parameterization is essential when designing hierarchical prior models. While a strong regularization result in prior dominating, poor regularization may lead to over-fitting. The Bayesian approach allows additional advantages over the classical methods such as uncertainty, but performance is in par with the common machine learning algorithms in terms of predictive power.

Our contributions can be summarized as follows:

- *Modeling contribution:* We present a sparse regression-based Bayesian accelerated failure time model where the sparsity is induced across predictors through a shrinkage prior.
- *Experimental contribution:* We identify relevant imaging-based prognostic factors that influence survival of patients with Glioblastoma. We demonstrate that inclusion of radiomic-based features enhance the predictive performance.

The type of model should be determined by the research objectives; if the goal is prediction, methods based on machine learning could be a better alternative. Except for the patient age the other clinical information such as medications or demographics (gender, race) were not available to be incorporated in the survival prediction model. Due to the absence of important factors relevant to the prediction task, survival task based on imaging data alone is very challenging, however, inclusion of radiomic-features in survival prediction showed a considerable improvement over the baseline.

## CHAPTER 6

### CONCLUSIONS AND FUTURE RESEARCH

Statistical Learning is based on a solid theoretical framework and mathematical principles. Deep learning on the hand, has emerged as a more pragmatic mechanism in the development of intelligent systems in real-world. Combining theory-driven statistical procedures and data-driven deep learning methods, in this study we addressed some of the key challenges encountered when handling real data. More specifically, class imbalance and confidence calibration when modeling rare events, estimating uncertainty when constructing models in safety-critical applications and modeling high dimensional data in time-to-event modeling. Here we provide concluding remarks and summarize open leads for future research.

*Class Imbalance Learning and confidence calibration:* Class imbalance presents one of the biggest difficulties in many real-world applications and devising algorithms that are adept at dealing with class imbalance is imperative. To this end, we proposed a cost-sensitive based approach which also leads to improved calibration. Differing from the methods centered around sampling-based approaches the proposed methods make maximum use of available data, considers the difficulty levels of individual samples and can be applied to any network without requiring any architectural modifications. It was theoretically justified and empirically validated on different application areas of cyber-security and healthcare to show the generalizability across domains. Future work will focus on applying the presented approach to a broader range of data sets with varying degrees of class imbalance, and performance comparisons under multiple criteria. In addition, studies on noisy and outlier samples in the minority group might prove an interesting area for further research. With regard to confidence calibration, factors affecting neural network miscalibration may constitute an interesting issue for future research to explore.

*Segmentation and Uncertainty Estimation:* The transition from scientific research to clinical practice depends on the trustworthiness of computer-aided diagnosis systems and raises profound questions about reliability, uncertainty, and robustness. For example, how to detect when the com-

putational algorithms get it wrong? Can we predict when the machine learning model fails, and how to make the algorithm robust to changes in real-world conditions and clinical data? These research questions are of central interest in safety-critical medical applications where accuracy is integral. Here we presented a stochastic variational inference-based approach for uncertainty estimation with the aim to identify the instances the algorithm is uncertain about. Performance evaluations on biomedical image segmentation data proved validity for the proposed approach. In the context of segmentation, there are several interesting research questions that needs further investigation. For instance, Deep network-based methods are prone to overfitting and hence evaluating the level of accuracy on new clinical data in the absence of ground truth is crucial in automated diagnosis. Methods are required to identify when the system fails and to evaluate the actual performance after implementation when a reference segmentation by an expert is inaccessible. Future efforts should further explore this issue before clinical adoption of automated systems. Moreover, the ultimate goal in computer-aided medical screening is to surpass human-level performance. Cancer histopathology reads in biomedical imaging requires expert knowledge, and it may subject to annotation bias while leading to a lack of agreement among human experts and therefore identifying the exact ground truth segmentation could be challenging. Synthesis of ground truth via adversarial learning appears to be a promising research direction in this context.

*Survival analysis and high dimensionality:* Due to the absence of important factors relevant to the prediction task, survival task based on imaging data alone is very challenging, however, inclusion of radiomic-features in survival prediction showed a considerable improvement over the baseline. We believe that integration of other important clinical information with imaging data will dramatically enhance the performance. Future research could explore the applicability of the methods developed here across other domains of time-to-event modeling. Future research should further experiment with other shrinkage priors that induces sparsity.

## REFERENCES

- [1] K Ruwani M Fernando and Chris P Tsokos. Deep and statistical learning in biomedical imaging: State of the art in 3d mri brain tumor segmentation. *arXiv preprint arXiv:2103.05529*, 2021.
- [2] © 2021 IEEE. Reprinted, with permission, from K Ruwani M Fernando and Chris P Tsokos. Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [3] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20–23, 2019.
- [4] Ashwin Carvalho, Stéphanie Lefèvre, Georg Schildbach, Jason Kong, and Francesco Borrelli. Automated driving: The role of forecasts and uncertainty—a control perspective. *European Journal of Control*, 24:14–32, 2015.
- [5] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.
- [6] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.
- [7] U Rajendra Acharya, Hamido Fujita, Oh Shu Lih, Yuki Hagiwara, Jen Hong Tan, and Muhammad Adam. Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network. *Information sciences*, 405:81–90, 2017.
- [8] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In

- 2012 *IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE, 2012.
- [9] Dan Li, Jianxin Zhang, Qiang Zhang, and Xiaopeng Wei. Classification of ecg signals based on 1d convolution neural network. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6. IEEE, 2017.
- [10] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.
- [11] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*, 2018.
- [12] Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.
- [13] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.
- [14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- [15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.
- [16] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161. Springer, 1998.
- [17] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [18] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*, 2018.

- [19] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42, 2018.
- [20] Zhi-Bo Zhu and Zhi-Huan Song. Fault diagnosis based on imbalance modified kernel fisher discriminant analysis. *Chemical Engineering Research and Design*, 88(8):936–951, 2010.
- [21] Zhenyu Wu, Yang Guo, Wenfang Lin, Shuyang Yu, and Yang Ji. A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. *Sensors*, 18(4):1096, 2018.
- [22] David A Cieslak, Nitesh V Chawla, and Aaron Striegel. Combating imbalance in network intrusion datasets. In *GrC*, pages 732–737, 2006.
- [23] M Emre Celebi, Hassan A Kingravi, Bakhtiyar Uddin, Hitoshi Iyatomi, Y Alp Aslandogan, William V Stoecker, and Randy H Moss. A methodological approach to the classification of dermoscopy images. *Computerized Medical imaging and graphics*, 31(6):362–373, 2007.
- [24] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2-3):427–436, 2008.
- [25] Matías 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] Masoumeh Zareapoor, Pourya Shamsolmoali, et al. Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia computer science*, 48(2015):679–685, 2015.
- [27] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.
- [28] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.
- [29] Byron C Wallace and Issa J Dahabreh. Improving class probability estimates for imbalanced data. *Knowledge and information systems*, 41(1):33–52, 2014.

- [30] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.
- [31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [32] Yuri Sousa Aurelio, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. Learning from imbalanced data sets with weighted cross-entropy function. *Neural Processing Letters*, 50(2):1937–1949, 2019.
- [33] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.
- [34] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pages 179–186. Nashville, USA, 1997.
- [35] 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.
- [36] 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.
- [37] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1):40–49, 2004.
- [38] Haibo He, Yang Bai, Eduardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.
- [39] 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.
- [40] 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.

- [41] Tomasz Maciejewski and Jerzy Stefanowski. Local neighbourhood extension of smote for mining imbalanced data. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 104–111. IEEE, 2011.
- [42] Cristiano L Castro and Antônio P Braga. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems*, 24(6):888–899, 2013.
- [43] 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.
- [44] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [45] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [46] Chris Seiffert, Taghi M Khoshgoftaar, and Jason Van Hulse. Improving software-quality predictions with data sampling and boosting. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(6):1283–1294, 2009.
- [47] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer, 2003.
- [48] 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.
- [49] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2011.
- [50] Xu-Ying Liu and Zhi-Hua Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 970–974. IEEE, 2006.



- [51] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.
- [52] 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.
- [53] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.
- [54] Chong Zhang, Kay Chen Tan, Haizhou Li, and Geok Soon Hong. A cost-sensitive deep belief network for imbalanced classification. *IEEE transactions on neural networks and learning systems*, 30(1):109–122, 2018.
- [55] 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.
- [56] 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 Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.
- [57] 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.
- [58] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.
- [59] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.
- [60] Gary King and Langche Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001.
- [61] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

- [62] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519, 2009.
- [63] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. *arXiv preprint arXiv:2002.09437*, 2020.
- [64] Mahdi Pakdaman Naeni, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access, 2015.
- [65] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.
- [66] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, pages 38–41, 2019.
- [67] Sireesha Rodda and Uma Shankar Rao Erothi. Class imbalance problem in the network intrusion detection systems. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2685–2688. IEEE, 2016.
- [68] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.
- [69] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.
- [70] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [71] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5:180161, 2018.

- [72] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kallou, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 168–172. IEEE, 2018.
- [73] Marc Combalia, Noel CF Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C Halpern, Susana Puig, et al. Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*, 2019.
- [74] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.
- [75] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT’2010*, pages 177–186. Springer, 2010.
- [76] Mingxing Tan. Google AI Blog: EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. <https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html>. [Online; accessed 2-October-2021].
- [77] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.
- [78] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.
- [79] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.
- [80] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.
- [81] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d

- cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.
- [82] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.
- [83] Onur Ozdemir, Benjamin Woodward, and Andrew A Berlin. Propagating uncertainty in multi-stage bayesian convolutional neural networks with application to pulmonary nodule detection. *arXiv preprint arXiv:1712.00497*, 2017.
- [84] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):1–14, 2017.
- [85] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, and Christian Wachinger. Inherent brain segmentation quality control from fully convnet monte carlo sampling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 664–672. Springer, 2018.
- [86] Oliver Dürr, Elvis Murina, Daniel Siegismund, Vasily Tolkachev, Stephan Steigele, and Beate Sick. Know when you don’t know: A robust deep learning approach in the presence of unknown phenotypes. *Assay and drug development technologies*, 16(6):343–349, 2018.
- [87] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.
- [88] Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59:101557, 2020.
- [89] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019.
- [90] Ian Osband. Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In *NIPS Workshop on Bayesian Deep Learning*, volume 192, 2016.

- [91] Simon AA Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus H Maier-Hein, SM Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. *arXiv preprint arXiv:1806.05034*, 2018.
- [92] Christian F Baumgartner, Kerem C Tezcan, Krishna Chaitanya, Andreas M Hötker, Urs J Muehlematter, Khoschy Schawkat, Anton S Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–127. Springer, 2019.
- [93] Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *arXiv preprint arXiv:2006.06015*, 2020.
- [94] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021.
- [95] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.
- [96] Sebastian Farquhar, Lewis Smith, and Yarin Gal. Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations. *arXiv preprint arXiv:2002.03704*, 2020.
- [97] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.
- [98] Christopher T Lloyd, Alessandro Sorichetta, and Andrew J Tatem. High resolution global gridded data for use in population studies. *Scientific data*, 4(1):1–17, 2017.
- [99] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki,

- et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.
- [100] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein. No new-net. In *International MICCAI Brainlesion Workshop*, pages 234–244. Springer, 2018.
- [101] 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.
- [102] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *International MICCAI brainlesion workshop*, pages 178–190. Springer, 2017.
- [103] Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320. Springer, 2018.
- [104] Zeyu Jiang, Changxing Ding, Minfeng Liu, and Dacheng Tao. Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In *International MICCAI Brainlesion Workshop*, pages 231–241. Springer, 2019.
- [105] Fabian Isensee, Paul F Jäger, Peter M Full, Philipp Vollmuth, and Klaus H Maier-Hein. Nnunet for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 118–132. Springer, 2020.
- [106] Hannelore K van der Burgh, Ruben Schmidt, Henk-Jan Westeneng, Marcel A de Reus, Leonard H van den Berg, and Martijn P van den Heuvel. Deep learning predictions of survival based on mri in amyotrophic lateral sclerosis. *NeuroImage: Clinical*, 13:361–369, 2017.
- [107] Luke Oakden-Rayner, Gustavo Carneiro, Taryn Bessen, Jacinto C Nascimento, Andrew P Bradley, and Lyle J Palmer. Precision radiology: predicting longevity using feature engineering and deep learning methods in a radiomics framework. *Scientific reports*, 7(1):1–13, 2017.

- [108] Sohi Bae, Yoon Seong Choi, Sung Soo Ahn, Jong Hee Chang, Seok-Gu Kang, Eui Hyun Kim, Se Hoon Kim, and Seung-Koo Lee. Radiomic mri phenotyping of glioblastoma: improving survival prediction. *Radiology*, 289(3):797–806, 2018.
- [109] Dong Nie, Han Zhang, Ehsan Adeli, Luyan Liu, and Dinggang Shen. 3d deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In *International conference on medical image computing and computer-assisted intervention*, pages 212–220. Springer, 2016.
- [110] Leon Weninger, Oliver Rippel, Simon Koppers, and Dorit Merhof. Segmentation of brain tumors and patient survival prediction: Methods for the brats 2018 challenge. In *International MICCAI brainlesion workshop*, pages 3–12. Springer, 2018.
- [111] Florian Kofler, Johannes C Paetzold, Ivan Ezhov, Suprosanna Shit, Daniel Krahulec, Jan S Kirschke, Claus Zimmer, Benedikt Wiestler, and Bjoern H Menze. A baseline for predicting glioblastoma patient survival time with classical statistical models and primitive features ignoring image information. In *International MICCAI Brainlesion Workshop*, pages 254–261. Springer, 2019.
- [112] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [113] Arnošt Komárek and Emmanuel Lesaffre. Bayesian accelerated failure time model with multivariate doubly interval-censored data and flexible distributional assumptions. *Journal of the American Statistical Association*, 103(482):523–533, 2008.
- [114] Kyu Ha Lee, Sounak Chakraborty, and Jianguo Sun. Bayesian variable selection in semi-parametric proportional hazards model for high dimensional survival data. *The International Journal of Biostatistics*, 7(1), 2011.
- [115] Kyu Ha Lee, Sounak Chakraborty, and Jianguo Sun. Survival prediction and variable selection with simultaneous shrinkage and grouping priors. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(2):114–127, 2015.
- [116] Ismaël Castillo, Johannes Schmidt-Hieber, and Aad Van der Vaart. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018, 2015.

- [117] Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR, 2009.
- [118] W Thomson, S Jabbari, AE Taylor, W Arlt, and DJ Smith. Simultaneous parameter estimation and variable selection via the logit-normal continuous analogue of the spike-and-slab prior. *Journal of the Royal Society Interface*, 16(150):20180572, 2019.
- [119] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.
- [120] Lee-Jen Wei. The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. *Statistics in medicine*, 11(14-15):1871–1879, 1992.
- [121] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.
- [122] Chris Hans. Model uncertainty and variable selection in bayesian lasso regression. *Statistics and Computing*, 20(2):221–229, 2010.
- [123] CBICA Image Processing Portal. A web accessible platform for imaging analytics; Center for Biomedical Image Computing and Analytics, University of Pennsylvania. <https://ipp.cbica.upenn.edu/>.
- [124] Joost JM Van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts. Computational radiomics system to decode the radiographic phenotype. *Cancer research*, 77(21):e104–e107, 2017.
- [125] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.
- [126] Avoid Infringement upon IEEE Copyright. <https://journals.ieeeauthorcenter.ieee.org/choose-a-publishing-agreement/avoid-infringement-upon-ieee-copyright/>. [Online; accessed 9-December-2021].



## APPENDIX A

### DESCRIPTION OF RADIOMIC FEATURES

Table A.1: Description of Radiomic Features Extracted from Pyradiomics

Feature Category	Name of the features
First Order Statistics	Energy, Total Energy, Entropy, Minimum, 10th percentile, 90th percentile, Maximum, Mean, Median, Interquartile Range, Range, Mean Absolute Deviation, Robust Mean Absolute Deviation, Root Mean Squared, Standard Deviation, Skewness, Kurtosis, Variance, Uniformity
Shape-based	Mesh volume, Voxel Volume, Surface Area, Surface Area to Volume Ratio, Sphericity, Compactness 1, Compactness 2, Spherical Disproportion, Maximum 3D diameter, Maximum 2D diameter (slice), Maximum 2D diameter (columns), Maximum 2D diameter (row), Major Axis Length, Minor Axis Length, Least Axis Length, Elongation, Flatness
Gray Level Cooccurrence Matrix (GLCM)	Autocorrelation, Joint Average, Cluster Prominence, Cluster shade, Cluster Tendency, Contrast, Correlation, Difference Average, Difference Entropy, Difference Variance, Joint Energy, Joint Entropy, Informational Measure of Correlation, Inverse Difference Moment, Maximal Correlation Coefficient, Inverse Difference Moment Normalized, Inverse Difference, Inverse Difference Normalized, Inverse Variance, Maximum Probability, Sum Average, Sum Entropy, Sum of Squares
Gray Level Run Length Matrix(GLRLM)	Short Run Emphasis, Long Run Emphasis, Gray Level Non-Uniformity, Gray Level Non-Uniformity Normalized, Run Length Non-Uniformity, Run Length Non-Uniformity Normalized, Run Percentage, Gray Level Variance, Run Variance, Run Entropy, Low Gray Level Run Emphasis, High Gray Level Run Emphasis, Short Run Low Gray Level Emphasis, Short Run High Gray Level Emphasis, Long Run Low Gray Level Emphasis, Long Run High Gray Level Emphasis
Gray Level Size Zone Matrix (GLSZM)	Small Area Emphasis, Large Area Emphasis, Gray Level Non-Uniformity, Gray Level Non-Uniformity Normalized, Size Zone Non-Uniformity, Size Zone Non-Uniformity Normalized, Zone Percentage, Gray Level Variance, Zone Variance, Zone Entropy, Low Gray Level Zone Emphasis, High Gray Level Zone Emphasis, Small Area Low Gray Level Emphasis, Small Area High Gray Level Emphasis, Large Area Low Gray Level Emphasis, Large Area High Gray Level Emphasis
Gray Level Dependent Matrix (GLDM)	Small Dependence Emphasis, Large Dependence Emphasis, Gray Level Non-Uniformity, Gray Level Non-Uniformity Normalized, Dependence Non-Uniformity, Dependence Non-Uniformity Normalized, Gray Level Variance, Dependence Variance, Dependence Entropy, Dependence Percentage, Low Gray Level Emphasis, High Gray Level Emphasis, Small Dependence Low Gray Level Emphasis, Small Dependence High Gray Level Emphasis, Large Dependence Low Gray Level Emphasis, Large Dependence High Gray Level Emphasis
Neighbouring Gray Tone Difference Matrix (NGTDM)	Coarseness, Contrast, Busyness, Complexity, Strength

**APPENDIX B**  
**COPYRIGHT PERMISSION**

**Case #01502128 - Copyright**

1 message

---

**customercare@copyright.com** <customercare@copyright.com>  
To: "ruwfernando@gmail.com" <ruwfernando@gmail.com>

Thu, Dec 9, 2021 at 1:51 PM

Dear Dr. Fernando:

It is nice to chat with you.

Please find attached the automatic grant to use your article or portions of it in your upcoming Dissertation.

Good luck.

Kind regards,

Jeanne

Jeanne Brewster  
Customer Account Specialist  
Copyright Clearance Center  
[222 Rosewood Drive](#)  
Danvers, MA 01923  
[www.copyright.com](#)  
Toll Free US +1.855.239.3415  
International +1.978-646-2600

[Facebook](#) - [Twitter](#) - [LinkedIn](#)


---


 **IEEE permission.pdf**  
247K

Figure B.1.: E-mail received from copyright clearance center (e-mail attachment is in Figure B.2.)

As mentioned in the attached statement in Figure B.2, "The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant" (depicted in Figure B.2). This grants permission to reuse the published article in the Dissertation. Also, in order to fulfill the IEEE Copyright requirements [126], the following copyright notice is included in the references: "©2021 IEEE. Reprinted, with permission, from K Ruwani M Fernando and Chris P Tsokos. Dynamically weighted balanced loss: class

imbalanced learning and confidence calibration of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2021.”

Home? Help✉ Email SupportK. Ruwani Fernando



Requesting permission to reuse content from an IEEE publication

**Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks**

Author: K. Ruwani M. Fernando  
Publication: IEEE Transactions on Neural Networks and Learning Systems  
Publisher: IEEE  
Date: Dec 31, 1969

Copyright © 1969, IEEE

**Thesis / Dissertation Reuse**

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications\\_standards/publications/rights/rights\\_link.html](http://www.ieee.org/publications_standards/publications/rights/rights_link.html) to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Figure B.2.: Permission grant statement.