University of Tübingen
Faculty of Science
Department of Computer Science

Master's Thesis Machine Learning

Robust and fast transformer-based models for bio-medical data

Lennart Metz

November 15, 2024

First Examiner

Dr. Katharina Eggensperger
AutoML for Science
Cluster of Excellence Machine Learning
University of Tübingen

Second Examiner

Prof. Dr. Nico Pfeifer
Methods in Medical Informatics
Department of Computer Science
University of Tübingen

Supervisor

Dr. Vadim Borisov
M3 Research Center
University Hospital Tübingen

Lennart Metz
Robust and fast transformer-based models for bio-medical data
Master’s Thesis Machine Learning
University of Tübingen
Editing period: 15/05/2024 – 15/11/2024

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum

Unterschrift

Abstract

Colorectal cancer (CRC) is among the most common types of cancer in both men and women. Survival rates greatly depend on how far into development the cancer is detected. Thus, early detection of CRC is a critical area of research. One possible avenue for detecting CRC is a patient's gut microbiome composition. The human gut microbiome is a complex system of billions of bacteria belonging to thousands of species and their composition can vary starkly among individuals. Furthermore, the interactions between the gut microbiome and human health are often complex. Consequently, machine learning (ML) is a valuable tool in the analysis of microbiome data. Most research applying ML models in the classification of gut microbiome data defaults to common models, such as logistic regression, random forest, and gradient-boosted trees, with little progress in developing more sophisticated, specialized approaches. In this thesis, I present MedPFN, a model designed for the classification of highly imbalanced gut microbiome data. MedPFN is a prior-data fitted network (PFN), a class of transformer models intended as general inference machines by being pretrained on large amounts of synthetic data. For this, I propose the Dirichlet-Mutinomial dataset prior, which generates synthetic classification tasks mimicking real bio-medical data. Furthermore, clinical data often presents the challenge of a highly imbalanced class distribution, as the majority of patients belong to the healthy cohort. Class imbalance poses a challenge for classification tasks, and I developed a curriculum training regime for PFNs with the goal of improving model performance on imbalanced datasets. I empirically study MedPFN, the DM prior and the curriculum training regime on a large clinical dataset that associates the microbiome composition of patients with the presence of CRC. MedPFN demonstrates an excellent performance and provides more reliable predictions than the considered baselines. By isolating the influence of the proposed methods, I find evidence that they enhance the model's predictive capabilities on the given dataset. Additionally, the training regime is shown to improve model performance regardless of dataset imbalance.

Contents

1 Introduction

The human microbiome is an integral part of the human body. While bacterial or fungal infections are a serious threat to human health, many areas in the body are teeming with microorganisms that are either harmless or even serve vital functions in the body. Some of these areas include the skin [Grice, 2011], mouth [Wade, 2013], and most importantly the gut [Cresci and Bawden,]. Inside the human digestive system, a plethora of different bacteria can be found in large quantities. Critically, the type of bacteria present and their abundances vary starkly between individuals. The human gut microbiome has long been known to aid the body in the digestion process. However, in recent years, research has shown the gut microbiome to have a strong connection to various other biological mechanisms [Cresci and Bawden,]. Several biological factors outside of the gut influence the composition of gut microbiota, including diseases, the environment and genetics [Dave et al., 2012]. Inversely, the microbiome affects other parts of the human body, including the central nervous system and the brain [Galland, 2014]. One of the most well-established connections is to the presence of colorectal cancer (CRC) in a patient [Rebersek, 2021]. CRC is among the most common types of cancer worldwide for both men and women and also the second largest cause of cancer-related deaths [Granados-Romero et al., 2017]. Early detection plays a vital role in reducing the mortality rate of CRC cases [Granados-Romero et al., 2017]. The gut microbiome is being researched as a pathway for early detection, treatment, prevention, and prognosis [Rebersek, 2021]. As the data collected on human microbiome profiles can be vast and their influence is often complex, machine learning (ML) is a powerful tool in this domain [Hernández Medina et al., 2022]. However, working on microbiome data and using it for the classifcation in a CRC diagnosis poses several challenges.

First, it is compositional in nature, which means it contains relative abundances instead of absolute values [Aitchison, 2018]. In other words, all features for one sample add up to one. Second, it is often sparse, as the absence of bacteria species in a patient leads to zero values in the abundance profile. Third, the target variable is often highly imbalanced in its distribution. The majority of patients belong to the healthy class, and only a small number are in the CRC cohort. The imbalance is a result of such diseases being rare within the population and the difficulty of acquiring more samples, which is intrinsic to the domain of medical data. Lastly, applying machine learning models to medical data with potential impacts on human health requires high levels of trust, and consequently, the interpretability of the decision-making process is crucial in this domain. The requirement applies to the diagnosis of CRC based on microbiome data, and any classifier constructed for this purpose has to consider the trustworthiness of its predictions.

Current research on classification tasks on microbiome data is often content with applying standard models such as tree-based classifiers, regression algorithms and support vector machines [Wu et al., 2023], [Papoutsoglou et al., 2023]. Studies on advancing the performance of ML models on microbiome data are typically focused on data preprocessing [Tolosana-Delgado et al., 2019], [Yerke et al., 2024], with a lack of research on developing more sophisticated approaches, such as deep learning-based methods. The lack of research leads to the motivation for this thesis, which is to develop methods intended for classification of highly imbalanced, compositional microbiome data, incorporating recent advances in deep neural networks. Specifically, I focus on prior-data fitted networks (PFN), which present a promising alternative to traditional approaches and other neural network architectures.

PFNs[Müller et al., 2021] are a novel approach that leverage the in-context learning abilities of transformers to perform one-shot training/inference. Unlike typical machine learning algorithms, they are not trained on a specific dataset or task. Instead, they are pretrained to solve many, synthetically generated classification problems with to learn a general ability of performing one-shot inference. This pretraining procedure is done once, and after completion, no further optimization has to be done. For an unseen dataset, the pretrained model is fed all training and test samples at once and outputs predictions for the target variable of all test samples in a single forward pass. The mechanism to produce informed predictions is the utilization of attention between data samples akin to in-context learning. The context here refers to the underlying relationships between features and the target variable as presented in the training data. The performance of PFNs' has been shown to be competitive with state-of-the-art (SOTA) models on various tasks [Hollmann et al., 2023], [den Breejen et al., 2024], [Picard and Ahmed, 2024]. Additionally, due to the described one-shot methodology, PFNs demonstrate very short training times, as training consists of only one forward pass. PFNs most significant application is in tabular classification, where they have demonstrated performances on par or even better than SOTA methods [Hollmann et al., 2023]. As predicting disease status from gut microbiome composition is such a tabular classification problem, PFNs present a promising alternative to common approaches like random forest, XGBoost, or logistic regresion.

Tree-based methods or regression approaches are prevalent in medical applications due to their intrinsic interpretability [Molnar et al., 2020a]. The feature weights from fitted regression models and the learned rule structure that make up a decision tree are a direct window into their decision-making processes. In comparison, deep learning methods function as black box models [Shwartz-Ziv and Tishby, 2017]. While they demonstrate exceptional performance on many tasks, their internal processes for arriving at the solutions they provide are opaque. This property is especially problematic in microbiome research, as interpretability is often an essential requirement for ML models, and the lack of interpretability is considered an obstacle in applying deep learning methods to microbiome data [Hernández Medina et al., 2022]. The field of interpretable machine learning (IML) has rapidly developed in machine learning research in recent years [Molnar et al., 2020b] and aims to provide model-agnostic methods that analyze the predictions provided by black box models in order to understand their decision-making. Furthermore, not only does understanding the decision-making process increase trustworthiness in itself, but it can also be corroborated by existing research in the application's domain. Additionally, IML can help discover novel research directions and is a useful tool to find patterns in complex data. The discussed motivations for IML are especially relevant in medical settings, as ML models can have a direct role in diagnosing diseases, designing treatments, and assessing risk for patients. Many of the most common IML methods require repeated retraining of models and other extensive computations when applied to large neural networks, and consequently, approximations are often needed [Lundberg and Lee, 2017]. PFNs, with their one-shot approach, are uniquely well adapted for several common IML methods, as outlined by [Rundel et al., 2024]. I aim to apply several IML methods to PFNs for microbiome data to investigate their decision-making process.

1.1 Problem statement, contributions and structure

To construct a PFN specialized in solving binary classification tasks on highly imbalanced, compositional, bio-medical data, several challenges need to be addressed.

- Dataset priors for tabular classification provided by previous research construct general tabular datasets [Müller et al., 2021], [Hollmann et al., 2023], since those PFNs are trained to solve classification problems for no explicit domain. Consequently, it is necessary to develop a method of generating synthetic datasets that imitate real, bio-medical data, to incorporate it into a dataset prior.
- Dataset imbalance complicates the training of classification models [Feng et al., 2021]. Common approaches to tackling this obstacle are not directly applicable due to the unique functionality of PFNs. This introduces the need for an approach adapted to the unique pretraining routine performed on PFNs.
- In order to assess the trustworthiness of the decision-making process, the interpretability of any model is of high importance and needs to be considered for any new approach.

With the research conducted in this thesis, I propose the following solutions to the stated problems. Firstly, a novel synthetic dataset prior, the **Dirichlet-Multinomial (DM)** prior, based on the idea of imitating the sequencing of real microbiome profiles. The DM prior generates valid compositional datasets combining the multinomial and Dirichlet distributions as a data sampling function. It incorporates the label-assigning procedure of randomly instantiated Bayesian Neural Networks (BNN) from [Hollmann et al., 2023], which yields complex relationships between the features and target variable. I show that the DM prior constructs datasets similar to a clinical dataset relating microbiome profiles of patients to a CRC diagnosis. Secondly, I put forward a **curriculum training regime** for the pretraining of PFNs. I designed it to improve model performance on highly imbalanced datasets by providing increasingly imbalanced supervised learning tasks and employing cost-sensitive training. Combining both methods, I present **MedPFN**, a PFN developed for the classification of imbalanced, compositional, bio-medical data. I empirically study

MedPFN on the clinical dataset and demonstrate an excellent performance compared to various baseline ML methods. The impact of the DM prior and training regime on the performance of MedPFN is explored in detail by isolating the effects of the proposed methods. I demonstrate the DM prior's ability improve the predictive quality of a PFN for the clinical classification task when compared to a dataset prior from previous research. I find empirical evidence, that the curriculum training regime enhances the model's ability of providing predictions in the setting of highly imbalanced datasets. Additionally, the experiments indicate that the training regime not only adapts the model to imbalanced datasets but incurs a general performance boost to the PFN regardless of the class distribution in the data. Lastly, I modify two IML methods for compositional data and explore the decision-making process of MedPFN in detail. This analysis shows that MedPFN utilizes known associations between the human gut microbiome and the presence of CRC.

To summarize, this thesis provides the following contributions:

- I designed and implemented the Dirichlet-Multinomial dataset prior for compositional, bio-medical data and empirically studied its effect on the performance of a specialized PFN.
- To enhance the quality of the predictions on imbalanced datasets, I developed a curriculum training regime for PFNs that is shown improve the performance of my model on a clinical microbiome dataset.
- I modified and applied several IML methods to understand the decision-making process of the developed model and gain insights into the biological factors connecting the human gut microbiome and colorectal cancer

This thesis is structured as follows. Chapter 2 outlines the relevant theoretical background for this thesis. First, I define the concepts of imbalanced datasets and compositional data. Additionally, the functionality of prior-data fitted networks and two interpretable machine learning approaches used in this thesis are described in detail. Chapter 3 explains the current state of research for the human gut microbiome, tabular classification, and PFNs. The developed methods of the Dirichlet-Multinomial prior, PFN training regime for imbalanced classification tasks and the modification of the IML methods for compositional data are detailed in Chapter 4. Chapter 5 lists the specifications of my experimental procedure. Results and subsequent analysis in Chapter 6 encompasses a closer inspection of the clinical dataset, comparison to data from the DM prior, and the performance of MedPFN compared to several baselines. Lastly, I discuss the results of the research conducted in this thesis chapter 7, including an outlook on potential avenues for future research.

2 Background

This chapter gives an overview of the essential theoretical background necessary for the proposed methods and conducted experiments. Firstly, I define imbalanced classification tasks and compositional data. Then, I describe the design and functionality of PFNs. Lastly, I specify two interpretable machine learning tasks that I adapt and apply in this research.

2.1 Imbalanced classification tasks

One challenge this thesis aims to address is the classification of imbalanced datasets, which is a binary supervised learning problem with a highly uneven class distribution in the data. A classification task consists of a set $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$ of N datapoints $\boldsymbol{x}^{(n)}$ with their corresponding label $y^{(n)}$. The task now comprises predicting the unknown label y_{new} for a new datapoint $\boldsymbol{x}_{new}$ given the dataset $\mathcal{D}$. A binary task restricts the class labels to two possible values, commonly defined as classes zero and one: $y \in \{0, 1\}$. A dataset presenting a classification problem is considered imbalanced if the number of samples of each class within the dataset is dissimilar. One measure to quantify the imbalance in a dataset is the imbalance ratio (IR) [Feng et al., 2021]:

$$\text{IR}(\mathcal{D}) = \frac{n_0}{n_0 + n_1} \tag{2.1}$$

Here n_0 and n_1 are the number of samples belonging to class zero and one. The convention is to denote the class with the majority of samples as class zero (majority class) and the other as class one (minority class). Consequently, the imbalance ratio is limited to IR $\in [0.5, 1]$. Imbalanced datasets pose two main challenges for classification tasks.

1. Training models naively on such datasets can generally lead to reduced performance [Ali et al., 2013]
   Let the classification risk [Vapnik, 1991] be defined as:

$$\begin{aligned} R(f) &= \mathbb{E}[\mathbb{1}(f(X) \neq Y)] \\ &= \text{Pr}(f(X) \neq Y) \\ &= \pi_0 \cdot \text{Pr}(f(X) \neq Y | Y = 0) + \pi_1 \cdot \text{Pr}(f(X) \neq Y | Y = 1) \end{aligned} \tag{2.2}$$

The classification risk is the expectation over a classifier f not predicting the correct label Y from data X or, equivalently, the probability of a classifier making an incorrect prediction. By the law of total probability, this can be decomposed into the sum of the probabilities of making a prediction error conditional on Y belonging to either class zero or class one [Feng et al., 2021]. π_0 and π_1 denote the probability of Y belonging to class zero or class one. Learning classification tasks is typically based on risk minimization:

$$f^* = \underset{f}{\text{argmin}} R(f) \tag{2.3}$$

Risk minimization is equivalent to reducing the number of incorrect predictions on the given classification task.

In an imbalanced classification problem, we know that $\pi_0 > \pi_1$. Thus, Equation 2.2 implies, that a simple way to reduce risk is to predict class zero more frequently than class one. The higher the class imbalance, the more effective this simple strategy is. As a result, training ML models on highly imbalanced classification tasks using regular risk minimization often leads to poor performance, especially in correctly classifying the minority class [Zou et al., 2016]. Additionally, in real-world applications, not identifying the rare minority class members is often considered more costly. For example, in the case of diagnosing the presence of CRC, misclassifying a patient with CRC as healthy has more severe consequences than misclassifying a healthy patient as having CRC.

There are several methods to adjust the training to take class imbalance into account. One such method is cost-sensitive learning [Feng et al., 2021]. Here, a cost is assigned to misclassifying each of the classes:

$$R(f) = C_0 \cdot \pi_0 \cdot \text{Pr}(f(X) \neq Y | Y = 0) + C_1 \cdot \pi_1 \cdot \text{Pr}(f(X) \neq Y | Y = 1) \tag{2.4}$$

The intention is to put increased weight on the minority class during risk minimization by setting its cost of misclassification higher than that of the majority class. Basic cost-sensitive learning is not applicable to PFNs, as no risk minimization is performed for an unseen dataset. It can only be incorporated into the pretraining phase, where it has to be modified to take the repeated sampling of different synthetic datasets into account. The curriculum training regime proposed in this thesis is based on cost-sensitive learning and is explained in detail in Section 4.2.

2. Measuring the performance of classification models in imbalanced data needs to be considered carefully. Accuracy measures the fraction of correct predictions in the total number of predictions made by the model:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2.5}$$

The majority class classifier is a naive predictor which predicts the majority class for all samples. This classifier's accuracy is exactly the IR (see Equation 2.1).

Thus, the higher the IR, the higher the accuracy of the uninformed majority class predictor. Accordingly, accuracy alone is an insufficient measure for evaluating the performance of a classifier on highly imbalanced datasets. Instead, additional metrics are considered in this thesis. The first metric is the area under the curve of the receiver operating characteristics (ROC AUC) [Huang and Ling, 2005]. The ROC is a curve plotting the true positive rate against the false positive rate for different classification threshold values.

The second metric is the F_1-score, which is the harmonic mean of the precision and recall:

$$\begin{aligned} F_1 &= 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \end{aligned} \tag{2.6}$$

Both metrics are considered more robust when evaluating imbalanced classification tasks [Jafarigol and Trafalis, 2023].

2.2 Compositional microbiome data

Compositional data is encountered in many scientific fields, including geology, chemistry, medicine, and microbiology. In short, features within a compositional dataset describe fractions of a total instead of absolute values. Meaning that for one sample all features sum up to a fixed number, usually one. Formally, samples from a compositional dataset with K features lie on the $K-1$-simplex [Aitchison, 2018]:

$$\sum_{k=1}^{K} \theta_k = 1 \quad \theta_k \in \mathbb{R},\ \theta_k \geq 0\ \forall k = 1, ..., K \tag{2.7}$$

The properties of compositional data entail complications for performing common statistical analyses. For example, the summation constraint leads to spurious correlations. When increasing one value, at least one other value must decrease. Consequently, at least one negative covariance is present in such a dataset [Pawlowsky-Glahn and Egozcue, 2006]. Examples of compositional datasets are the molecule concentration in a solution, genetic phenotypes within a population, and the microbiome composition in the human gut. The DM prior is motivated by the real generating mechanism behind gut microbiome data. The mechanism is as follows:

We have a group of patients with a diverse and unique set of bacteria in their gut. There are K different species of bacteria considered. The patients differ in which species exist in their individual gut microbiome and their respective composition. This means that each

patient only carries a subset of all possible bacteria in their digestive tract, and additionally, the frequency of bacteria species varies over individuals. The data expressing a patient's unique gut microbiome profile are the relative abundances, which assign each species a fraction of the total number of bacteria present in that individual. Note that species not carried by a patient have an abundance of zero. Furthermore, relative abundances are a compositional dataset as defined in Equation 2.7.
In practice, it is impossible to measure all bacteria in a patient, and the gut microbiome abundances are not read of directly but rather measured using a tissue or fecal sample. This is done via different DNA or RNA sequencing methods [Allali et al., 2017] that identify species by their unique genetic profile. For a given sample, M bacteria are sequenced and mapped to a strain, and the final counts are ultimately normalized by M. Typically, only a random subset of the bacteria in a sample are measured. Consequently, the final gut microbiome profile is not only an indirect measurement but additionally, an approximation of a sample. For N patient the gut microbiome data can be expressed as $\boldsymbol{X} \in \mathbb{R}^{N \times K}$.
The sampling procedure can be described as a draw from a multinomial distribution $\mathbf{Multi}(\boldsymbol{x}|M, \boldsymbol{\theta})$, where the probabilities are defined by the true abundances of bacteria in a patient: $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$. The set of true abundances within a patient can be interpreted as being drawn from a distribution over possible abundance profiles. The Dirichlet distribution is well suited for this, as it is defined over sets of values that fulfill the compositionality requirement from Equation 2.7. The connection between the multinomial distribution, the Dirichlet distribution, and gut microbiome composition is frequently used to model microbiome data [Holmes et al., 2012], [Chen and Li, 2013], [Wadsworth et al., 2017]. This relationship is the motivation for the proposed Dirichlet-Multinomial prior and will be described in more detail in Section 4.1.

2.3 Prior-data fitted networks

Prior-data fitted networks (PFNs), introduced by [Müller et al., 2021], are a novel approach to solving supervised learning problems. Inspired by the recent breakthroughs of large language models [Dong et al., 2024], they use in-context learning to approximate the posterior distribution for a given classification task. These models learn to solve inference problems generally, and a trained model is not restricted to a single task. They are pretrained on large amounts of synthetic datasets and can then be applied out-of-the-box on unseen tasks without any additional training in the usual sense. This leads to an immense speedup compared to state-of-the-art models [Müller et al., 2021], [Hollmann et al., 2023]. However, these models are currently limited to relatively small datasets. The following sections detail the theoretical foundation of PFNs, their practical implementation, including notable variations, and their current capabilities.

2.3.1 Theoretical foundation

Prior-data fitted networks aim to approximate Bayesian inference for supervised learning problems. From a Bayesian perspective, we would like to learn the posterior predictive distribution (PPD) $p(y|\boldsymbol{x}, \mathcal{D})$ that describes the probability distribution over possible values of y, given the datapoint $\boldsymbol{x}$ and the dataset $\mathcal{D}$. The PPD can be inferred as:

$$\begin{aligned} p(y|\boldsymbol{x}, \mathcal{D}) &= \int_{\mathcal{H}} p(y|\boldsymbol{x}, \mathcal{D}, h) dh \\ &= \int_{\mathcal{H}} p(y|\boldsymbol{x}, h) p(h|\mathcal{D}) dh \\ &= \int_{\mathcal{H}} p(y|\boldsymbol{x}, h) \left[\frac{p(\mathcal{D}|h)p(h)}{\int p(\mathcal{D}|h)p(h)dh} \right] dh \\ &\propto \int_{\mathcal{H}} p(y|\boldsymbol{x}, h) p(\mathcal{D}|h) p(h) dh \end{aligned} \tag{2.8}$$

Here h denotes a hypothesis or data-generating mechanism. This is the underlying process that encapsulates the relationships between data and labels within the dataset $\mathcal{D}$. $\mathcal{H}$ is the space over all hypotheses and can be viewed as a prior that determines the probability of an individual hypothesis. The PPD is approximated by integrating over this space of hypotheses, with $p(\mathcal{D}|h)$ being the likelihood of the dataset given the hypothesis h. In the case of the microbiome dataset this would be the biological and environmental factors that influence the microbiome composition of a patient's gut and the development of cancer.

PFNs are defined as a parameterized model of the PPD $q_\theta(y|\boldsymbol{x}, \mathcal{D})$, that defines a probability distribution over the possible values of y given the dataset $\mathcal{D}$ and novel sample $\boldsymbol{x}$. The authors of [Müller et al., 2021] define the following loss for PFNs, the *Prior-Data Negative Log-Likelihood (Prior-Data NLL)*:

$$l_\theta = \mathbb{E}_{\mathcal{D} \cup \{x,y\} \sim p(\mathcal{D})} [-\log q_\theta(y|x, \mathcal{D})] \tag{2.9}$$

In words, minimizing this loss corresponds to maximizing the predicted probability of the approximated PPD for the correct label in expectation over datasets drawn from the probability distribution over datasets.

2.3.2 Practical implementation

The practical implementation of PFNs as presented by [Müller et al., 2021] can be divided into three parts: The construction of the **training regime**, the **dataset prior**, and the **architecture**.

2.3.2.1 Training regime

The goal of the training is to fit the model as an approximation of the PPD. The training regime is shown in Algorithm 1. Given a prior over synthetic datasets $p(\mathcal{D})$, we draw a dataset of N training samples and M test samples. The model is then optimized by performing gradient descent on the stochastic loss approximation of the *Prior-Data NLL*: $\bar{\ell}_\theta = \sum_{m=1}^{M}(-\log q_\theta(y^{(m)}|\boldsymbol{x}^{(m)}, \mathcal{D}_{train}))$. As [Müller et al., 2021] have shown, the *Prior-Data NLL* is equivalent to the expectation of the cross-entropy between the PPD and its approximation q_θ. Consequently, an update step is realized by computing the cross-entropy-loss between the predictions and the correct labels. We then optimize the parameters of the model via stochastic gradient descent. These steps are repeated for E datasets.

Algorithm 1: Training a PFN model by Fitting Prior-Data (From [Müller et al., 2021])

```
Input  : A prior distribution over datasets $p(\mathcal{D})$, from which datasets can be drawn
         and the number $E$ of datasets to fit the model on.
Output : A model $q_\theta$ that will approximate the PPD
Initialize the neural network $q_\theta$;
for $j \leftarrow 1$ to $E$ do
    Sample $\mathcal{D}_{train} \cup \mathcal{D}_{test} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N} \cup \{(\boldsymbol{x}^{(m)}, y^{(m)})\}_{m=1}^{M} \sim p(\mathcal{D})$;
    Compute stochastic loss approx. $\bar{\ell}_\theta = \sum_{m=1}^{M}(-\log q_\theta(y^{(m)}|\boldsymbol{x}^{(m)}, \mathcal{D}_{train}))$;
    Update parameters $\theta$ with stochastic gradient descent on $\nabla_\theta \bar{\ell}_\theta$;
end
```

2.3.2.2 Dataset prior

The realization of the dataset prior is implemented as a two-step sampling mechanism $p(\mathcal{D}) = \mathbb{E}_{h \sim p(h)}[p(\mathcal{D}|h)]$. First, a hypothesis is drawn from a prior over hypotheses and then a dataset is sampled from this hypothesis. Practically, this means we need a method of generating hypotheses, from which we can draw samples. For illustrative purposes, this simple example is given to explain how this dataset prior is meant to function:

Assume we have two distributions $p(\mu)$ with $\mu \in \mathbb{R}^n$ and $p(\Sigma)$ with $\Sigma \in \mathbb{R}^{n \times n}$, and a set of functions F with $f \in F : \mathbb{R}^n \to \mathbb{R}$. We sample a mean μ_h and a covariance matrix Σ_h to define a normal distribution as $\mathcal{N}(\mu_h, \Sigma_h)$ and draw a function f_h at random. These two compose the hypothesis or data generating mechanism $\{f_h, \mathcal{N}(\mu_h, \Sigma_h)\}$. From the normal distribution, we can now draw individual datapoints $x \sim \mathcal{N}(\mu_h, \Sigma_h)$ and get the corresponding labels by evaluating $f_h(x)$.

When designing a concrete dataset prior, we consider the PPD the PFN is intended to approximate. There is no specific methodology or objective criteria for the construction of the prior presented in [Müller et al., 2021]. The authors of [Müller et al., 2021], as well as subsequent research into training PFNs ([Hollmann et al., 2023], [den Breejen et al., 2024]), propose priors that generally follow two principles. (1) *The prior should generate datasets with similar properties as the data encountered in the tasks the model is intended for* and (2) *The prior should generate diverse datasets*. Although the research does not provide clear evidence of this, intuitively, it seems sensible that the more similar the datasets seen during training are to an unseen dataset we wish to predict, the better the PPD approximation for this dataset is likely to be. The prior generating diverse datasets is intended to drive the model to learn general ways of inference instead of abusing simple patterns in the data akin to overfitting in regular neural net setups. Furthermore, PFNs that are designed to be applied to a broader range of scenarios need more varied datesets being generated by the prior.

2.3.2.3 Architecture

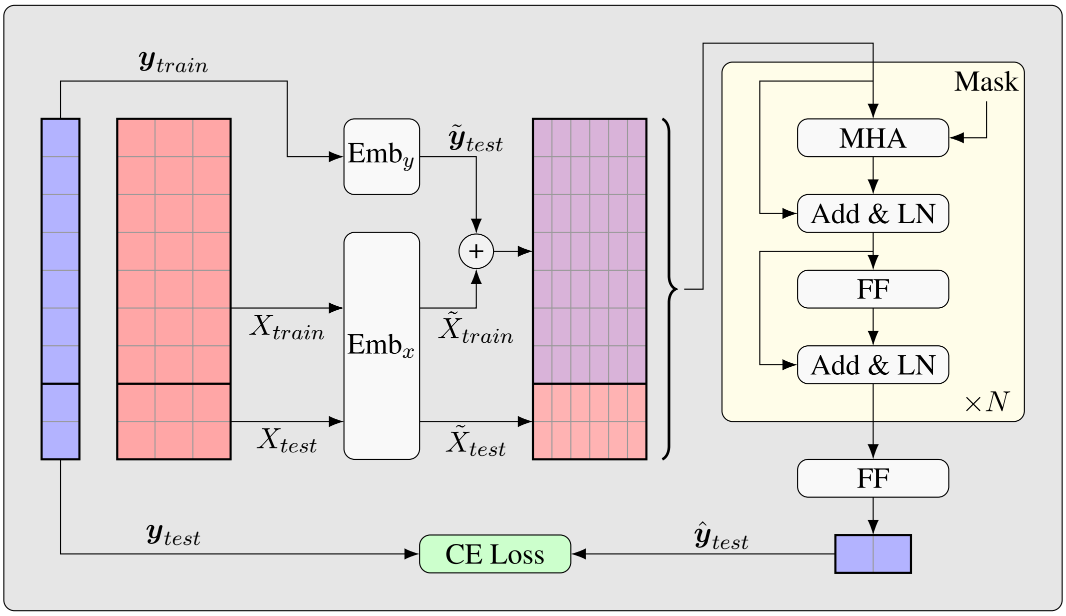


Figure 2.1: Diagram of the PFN architecture. Data and training labels are mapped to an embedding. Training labels are added to the training data and used as input together with the test data. The main model consists of N standard encoder layers in sequence. Output is then mapped to probabilities via a final feed-forward layer. Output and target labels are used in cross-entropy loss for optimization of the model parameters.

Prior-data fitted networks can, in theory, be implemented using different architectures. However, in practice transformers are the architecture of choice. Large language models have shown the capability for in-context learning, which is the desired behavior for one-shot classification on a novel dataset. Additionally, the transformer architecture has the distinct advantage of working on sequences of different lengths, which in the context of PFNs makes it able to work on datasets of different sizes without any preprocessing. The research for this thesis is based on the specific PFN transformer architecture as proposed

in [Hollmann et al., 2023] and will be explained in detail in this section.
The model architecture is visualized in Figure 2.1. The input for the model is an entire dataset

$$\mathcal{D} = \mathcal{D}_{train} \cup \boldsymbol{y}_{test} = (X_{train}, \boldsymbol{y}_{train}) \cup \boldsymbol{y}_{test}$$

for a single forward pass. All datapoints are mapped to an embedding via the data-encoder Emb_x. Likewise, the labels of the training samples are embedded via a label-encoder Emb_y. Both of these encoders are a single linear layer. The training label $\tilde{\boldsymbol{y}}_{test}$ embeddings are then added to the training data embeddings $\tilde{X}_{train}$, concatenated to the test sample embeddings $\tilde{X}_{test}$ and passed on. Importantly, the addition of the label embedding is only done for the training samples and not for the test samples. The test labels $\tilde{\boldsymbol{y}}_{test}$ are only used for the calculation of the cross-entropy loss between the predicted and target labels.
The central part of the model is a standard encoder-only transformer. The transformer [Vaswani et al., 2017] consists of N identical encoder layers. An encoder layer receives the input and uses this as the query, key, and value in a multi-head attention (MHA) block. A residual from the input is added to the output of the MHA. Afterwards, a layer normalization (LN) [Ba et al., 2016] is applied. The result is fed pointwise into a 2-layer feed-forward neural network with GeLU activations [Hendrycks and Gimpel, 2016] and dropout after each layer. In the next step, the output from the previous layer normalization is added as a residual to the output of the feed-forward layer as a residual, and a second layer normalization is applied. The data and label embedding layers are also learned during training. Furthermore, an attention mask is supplied to the MHSA that allows all training and test samples to attend the training samples but blocks any sample from attending any of the test samples. The mask for a trainset of size four and a testset of size two would look like this:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{2.10}$$

It should be noted that while test samples only attend to themselves, their information is not ignored but transmitted to the next segment via the residual connection. As a last step in the model, transformer's output is fed into a 2-layer feed-forward network with a GeLU activation after the first layer, which maps to a final output consisting of the predicted class probabilities. Finally, the loss is the cross-entropy between the test labels and the predicted probabilities. Optimization is done by performing gradient descent of this loss to all parameters in the model. Optimization includes all layers in the transformers, the final decoding layer and the two embedding layers.

2.3.3 Ensembling and finetuning

There are two additions that can improve the performance of PFNs. Ensembling as introduced by [Hollmann et al., 2023] and finetuning from [den Breejen et al., 2024]. For ensembling, N_{ens} modified training instances are created. This is done by randomly rotating feature positions and uniformly inverting the labels. Additionally, half of modified instances are scaled using a power transformation per feature. The model is passed the modified training sets separately. Then, the N_{ens} outputs are averaged into a single prediction. The experimental results by [Hollmann et al., 2023] suggest this increases the predictive accuracy for their PFN.

The finetuning procedure performs a small amount of weight updates to the network based on the training data. For this, the training data is randomly split into a subset of training and validation data for each iteration. [den Breejen et al., 2024] propose a small number of gradient steps with a static learning rate of 10^{-5}.

2.4 Interpretable machine learning for PFNs

This section details two popular interpretable machine learning (IML) methods that I adapted for use on compositional datasets.

2.4.1 Feature importance

Feature importance (FI) techniques measure the impact an individual feature has on the predictive performance of a machine learning model. One popular approach is the leave-one-covariate-out (LOCO) method. For this, the ML model is trained on the full dataset, and the performance measured as a baseline. Then, one feature is removed, the model is retrained on the reduced dataset, and the performance is compared to the baseline. Formally, the LOCO values for one feature can be defined as:

$$\text{LOCO}_k = S_{ML}\left(\{\boldsymbol{x}_{K\setminus k}^{(n)}\}_{n=1}^{N}, \boldsymbol{y}\right) - S_{ML}\left(\{\boldsymbol{x}_{K}^{(n)}\}_{n=1}^{N}, \boldsymbol{y}\right) \tag{2.11}$$

S_{ML} is the score the ML model has on an arbitrary metric for the given dataset. In the case of this thesis, the ROC AUC and F_1-score of a regular full cross-validation run are measured.

2.4.2 Feature effect

Feature effect methods (FE) measure the change in predicted probabilities of a model when perturbing a feature. One such method is individual conditional expectation (ICE) curves. ICE curves illustrate the predicted class probabilities of one sample for different values of an individual feature k. To this end, a space of T different values for feature k is defined: $\kappa = \{\tilde{k}_1, ..., \tilde{k}_T\}$. Given one test sample m, we measure the probabilities P predicted by the trained model when setting k to all values defined in κ. The other features remain fixed during this procedure. Furthermore, the feature change is only applied to the test data, the model is trained once on the original, unperturbed training data and not changed afterward. ICE curves probe what effect a particular feature has on the trained model. For feature k, each value t in the ICE curve of patient m is defined as:

$$\phi_{k,t}^{(m)} = P_{PFN}\left(\tilde{\boldsymbol{x}}_t^{(m)}\right) \quad \text{with} \quad \tilde{x}_{j,t}^{(m)} = \begin{cases} \tilde{k}_t & \text{if } j = k \\ x_j^{(m)} & \text{else} \end{cases} \tag{2.12}$$

Evaluating this for $t = 1, ..., T$ produces an ICE curve of the form:

$$\text{ICE}_k^{(m)} = \{(\tilde{k}_1, ..., \tilde{k}_T), (\phi_{k,1}^{(m)}, ..., \phi_{k,T}^{(m)})\} \tag{2.13}$$

ICE curves are constructed for individual test samples. To increase the amount of information displayed, plots include ICE curves for many test samples. Another way to condense the information from M test samples are partial dependence plots (PDP). These average the change in predicted probabilities over all test samples:

$$\text{PDP}_k = \{(\tilde{k}_1, ..., \tilde{k}_T), (\Psi_{k,1}, ..., \Psi_{k,T})\} \quad \text{with} \quad \Psi_{k,t} = \frac{1}{M}\sum_{m=1}^{M} \phi_{k,t}^{(m)} \tag{2.14}$$

Both methods can impose a high runtime cost. LOCO requires retraining of the model for each feature that is left out to measure the effect of its absence. ICE curves are constructed through repeated evaluation of the model on perturbed data samples, which ideally should be done for multiple samples and various features. For complex models that come with a sizable runtime cost in training and evaluation, like deep neural networks, these methods are often cumbersome. As shown by [Rundel et al., 2024], PFNs are uniquely well adapted to execute LOCO analysis and generate ICE plots. The retraining required for each LOCO value is a simple forward pass. Similarly, ICE curves at different perturbed values for one feature can be simultaneously evaluated on many test samples as they are passed into a PFN at once. My IML analysis is based on the research by [Rundel et al., 2024] and adapted to compositional data. It is further discussed in Section 4.3.

3 Related research

This section explores the current state of research in machine learning for microbiome applications, tabular classification and the recently developed prior-data fitted networks.

3.1 The human gut microbiome and machine learning

The human gut microbiome is a complex system interacting with a plethora of bodily functions even outside of the digestive tract. Research in recent years has found a seemingly endless interaction with other aspects of human health. The gut microbiome has been associated with the functioning of the immune system [Kau et al., 2011], including chronic inflammation [Rizzetto et al., 2018], liver disease [Rui Wang, 2021] and HIV infections [Liu et al., 2017]. Furthermore, it has been associated with diabtetes [Gou et al., 2020], brain health [Galland, 2014], and even mental disorders, such as depression and anxiety [Peirce and Alviña, 2019]. A more intuitive connection is between the gut microbiome and inflammatory bowel disease [Halfvarson et al., 2017] and colorectal cancer [**?**]. The full extent of the relationship between the gut microbiome and the human body is still far from fully understood. As humans have thousands of different bacteria in their digestive system and the microbiotic makeup varies starkly between individuals, the data presented here is highly complex. Machine learning is a useful tool in unraveling the biological mechanisms of the gut microbiome. It can be used to visualize and cluster microbiome data through unsupervised learning [Li et al., 2022], applied disease prediction [Manandhar et al., 2021], and identifying associations between bacteria and diseases [Liu et al., 2022]. ML models have been used for the prediction of CRC from microbiome data in multiple studies. [Wu et al., 2023] apply LASSO regression to classify healthy and diseased patients. [Konishi et al., 2022] utilize the H2O AutoML framework [LeDell and Poirier, 2020] for the same purpose. A comparison of ML approaches on classifying multiple diseases from microbiome data was done by [Su et al., 2022]. They measure the classification performance of random forest, K-nearest neighbors, multi-layer perceptron, support vector machine, and a graph convolutional neural network, concluding that random forest outperforms the rest. More research employing ML models for microbiome data classification problems can be found, but there is a distinct lack of methods purposely developed for this task or a uniform benchmark.

3.2 Tabular classification

Microbiome datasets consist of a vector of real values per patient that encodes the abundances of a fixed number of bacteria. As such, this is considered a tabular dataset. Tabular datasets are still among the most common types of data encountered in the real world. While deep learning approaches have revolutionized and subsequently dominated the field of machine learning in many areas, such as convolutional neural networks for computer vision and transformer networks for natural language applications, research on deep learning for tabular classification problems has not seen similar breakthroughs. An extensive study and proposal of a benchmark for tabular classification [Borisov et al., 2021] showed that tree-based models, such as XGBoost [Chen and Guestrin, 2016], CatBoost [Prokhorenkova et al., 2018] and Random Forest [Breiman, 2001], outperformed the state-of-the-art deep learning approaches like SAINT [Somepalli et al., 2021] or NODE [Popov et al., 2019]. A subsequent analysis of this issue [Grinsztajn et al., 2022] confirmed these results and attempted to find empirical evidence for pathological shortcomings of neural networks on tabular data. They propose several factors that favor tree-based methods over neural networks. Mainly, non-smooth target functions that smooth approximators like neural networks struggle with and features that are often uninformative and non rotationally-invariant. Recently, a new class of neural network-based models have demonstrated to reliably beat traditional tree-based approaches. Prior-data fitted networks are a novel approach at solving learning problems based on transformer architecture. As they are the central focus of this thesis, the state of the current research on PFNs is discussed in detail in the next section.

3.3 Prior-data fitted networks

This section outlines the current state of research on PFNs, including dataset priors, capabilities, and limitations.

3.3.1 Notable dataset priors

Prior-data fitted networks were initially proposed by [Müller et al., 2021]. They show PFNs' ability to approximate Gaussian Processes by constructing a dataset prior using a multivariate normal distribution and extend this approach to a variant that incorporates a prior over hyperparameters as commonly used in GPs [Williams and Rasmussen, 2006]. Furthermore, they build a dataset prior that instantiates a Bayesian Neural Network (BNN) from a distribution over model architectures and weights and use that BNN to label randomly sampled data points. Lastly, [Müller et al., 2021] present a dataset prior for handwritten digits and letters that consists of randomly drawn straight lines. They demonstrate a strong performance of the different PFNs they trained on the respective dataset priors. Most notably, the BNN prior was used to train a model for tabular datasets.

[Hollmann et al., 2023] expand on the approach of PFNs for general tabular classification problems. They propose a dataset prior that is conceptually based on structural causal models (SCMs) [Pearl, 2000]. In short, this prior instantiates directed acyclic graphs with noisy and sparse connections between nodes. Assigning features and a target variable to individual notes, a dataset is constructed by repeatedly sampling noise and evaluating the graph. This is intended to simulate the complex causal structure between features and a target variable. The SCM prior is implemented through BNNs, where each input, hidden neuron, and output serve as nodes within the imagined graph. The continuous values of the target variable are discretized by binning with random boundaries. The difference between the SCM prior and BNN prior is the parts of the network to which the features and target can be assigned. The BNN prior restricts features to the inputs and the target to be the output neuron, whereas the SCM prior allows features and the target to be anywhere within the network. The DM prior proposed in this thesis is based on the BNN prior, and this process will be explained in detail in Section 4.1. While the research of [Hollmann et al., 2023] suggests the SCM prior leads to improved results compared to the BNN prior, it is not directly possible to modify the DM dataset prior to generate the imitation bio-medical data as desired. The reason for this is the assignment of features to hidden neurons, which complicates any intervention that aims to modify the feature distribution.

[den Breejen et al., 2024] propose the forest prior, which uses decision trees to generate datasets to teach a PFN general causal inference on tabular classification tasks. They motivate this dataset prior by wanting to generate complex datasets, as opposed to imitating real-life datasets as [Hollmann et al., 2023] suggest. For this, they sample data and continuous labels from normal distributions. Then, a decision tree regressor is fit on the data and labels. A second set of data is sampled and assigned labels through the previously fitted decision tree regressor. The first set of data and labels is discarded and the second set is the output of the forest prior. While the data sampling could be substituted such that it generates bio-medical data, the continuous labels generated by the regressor are not as evenly distributed as the BNN labels (see Figure 4.3). As a consequence, they cannot be binned as flexibly as necessary for the training regime described in Section 4.2.

3.3.2 Capabilities and limitations

PFNs have shown remarkable performance in various domains. [Müller et al., 2021] have shown that PFNs can be used to approximate the PPD of Gaussian Processes accurately. Furthermore, they demonstrated PFNs' ability to solve a small computer vision task and perform well on tabular classification problems. [Hollmann et al., 2023] introduced TabPFN, an out-of-the-box tabular classifier. They show that TabPFN outperforms SOTA gradient boosted trees methods, such as CatBoost [Prokhorenkova et al., 2018] and XGBoost [Chen and Guestrin, 2016], and the AutoML methods ASKL2.0 and Autogluon on a suite of tabular classification tasks sourced from OpenML [Vanschoren et al., 2013]. Not only does TabPFN rank the best on all performance metrics, but it has significantly shorter training times than the competition. This is the results of the one-shot classification setup of pretrained PFNs. Several studies have confirmed the compelling performance of TabPFN [den Breejen et al., 2024], [McElfresh et al., 2023], [Picard and Ahmed, 2024]. Furthermore, the proposed forest prior and finetuning procedure from [den Breejen et al., 2024] slightly improves upon the TabPFN model according to their experiments.

While exhibiting impressive performance, the architecture and approach of PFNs does come with some limitations. While TabPFN is capable of handling categorical features and NaNs, the model can only handle up to 100 features. For higher dimensional datasets, some feature selection technique or dimensionality reduction procedure has to be utilized. Similarly, TabPFN can only be applied to tasks with up to 10 classes, but PFNs theoretically have no restriction in this aspect. As PFNs are transformers, they can handle input sequences of arbitrary length. However, the complexity of the multi-head attention blocks in the transformer architecture is quadratic in the input length $O(n^2)$. Consequently, the computational feasibility of PFNs is somewhat restricted in the number of samples in a dataset. TabPFN is trained on synthetic datasets with only 1024 samples. While the model still shows surprisingly strong performance on datasets with up to 5000 samples, much larger datasets than that bring a rapidly increasing cost in compute and memory requirements with them, in addition to unreliable performance. Another point of concern is the approach of training being inference. Meaning that a novel dataset is "learned" in one single forward pass. While this does entail exceptionally short training times, the flip side is that inference is exceedingly expensive and slow compared to other classification approaches. [Müller et al., 2023] propose a method that instantiates small, trained feed-forward neural networks through PFNs that decouples training and inference. While this cuts down the inference time significantly, it also deteriorates the performance. Lastly, PFN models are typically quite large. TabPFN consists of 25.82M parameters, which imposes a memory footprint exceeding other models with competitive performance on small, tabular datasets.

4 Methodology

This chapter outlines the methods developed in the thesis. It describes the Dirichlet-Multinomial prior that was developed for compositional bio-medical data with a binary classification task. Also, the specialized training regime for PFNs for highly imbalanced binary datasets is detailed. Lastly, I explain some minor modifications to two interpretable machine learning methods for compositional data.

4.1 Synthetic dataset prior for compositional bio-medical data

In this section, I define the Dirichlet-Multinomial prior intended to generate synthetic datasets mimicking real bio-medical data. The DM-prior is defined formally, and its design is motivated by connecting it to the underlying process of how the compositional microbiome dataset is generated in the real world. The DM-prior is outlined in Algorithm 2 and visualized in Figure 4.2. Given a set of inputs $(a_1, a_2, b_1, b_2, N, M, K, \mathcal{F})$ the DM-prior generates unique datasets $\{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$ with $\boldsymbol{x}^{(n)} \in \mathbb{R}^K$ and $y^{(n)} \in \mathbb{R}$. Here, N corresponds to the number of patients/samples, K is the number of bacteria/features, (a_1, a_2, b_1, b_2, M) are parameters that regulate the data generating process, and $\mathcal{F}$ is a distribution over Bayesian Neural Network (BNN) architectures and weights, that map data samples to a label: $f \in \mathcal{F} : f(\boldsymbol{x}^{(n)}) = y^{(n)}$.

The DM prior can be split into two separate components, the data generating mechanism and mapping data to labels.

4.1.1 Data generating mechanism

The method of generating data that mimics compositional, bio-medical data is inspired by the medical process of human gut microbiome profiling as described in 2.2. It encompasses the first two loops in Algorithm 2. For a more intuitive understanding of the design choices, I will begin by detailing the second for loop and its motivation.

Let $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$ be the relative abundances of the K different species of bacteria in a patients gut. As Abundances they entail the properties $\theta_k \in [0, 1]$ and $\sum_{k=1}^{K} = 1$, which conveniently also lets the abundances be interpreted as probabilities. Next, we define $\hat{\boldsymbol{x}}^{(n)} = (\hat{x}_1, ..., \hat{x}_K)$ as the initial output from a sequencing method applied to one patient's microbiome sample. These are the total counts of bacteria found in the sample as divided into their respective species. Then, the probability of a sequencing result of M total bacteria from a patient's sample can be described using the multinomial distribution,

Algorithm 2: Data sampling process of the Dirichlet-Multinomial prior.

```
Input  : Parameters $(a_1, a_2, b_1, b_2, M)$, # samples $N$, # features $K$ and distribution
         over BNNs $\mathcal{F}$
Output : Synthetic data of $N$ samples $\boldsymbol{x}^n \in \mathbb{R}^K$ with $K$ features
$f \sim \mathcal{F}$
for $k \leftarrow 1$ to $K$ do
    $\beta_k^{(1)} \sim \mathcal{U}_1(a_1, b_1)$
    $\beta_k^{(2)} \sim \mathcal{U}_2(a_2, b_2)$
    $\alpha_k \sim \text{Beta}(\beta_k^{(1)}, \beta_k^{(2)})$
end                                                        } Data generation
for $n \leftarrow 1$ to $N$ do
    $\boldsymbol{\theta}^{(n)} \sim \text{Dir}(\alpha_1, ..., \alpha_K)$
    $\hat{\boldsymbol{x}}^{(n)} \sim \text{Multi}(M, \boldsymbol{\theta}^{(n)})$
    $\boldsymbol{x}^{(n)} = \frac{1}{M} \cdot \hat{\boldsymbol{x}}^{(n)}$
end
for $n \leftarrow 1$ to $N$ do
    $y^{(n)} = f(\boldsymbol{x}^{(n)})$                     } Mapping to labels
end
```

where the necessary probabilities $\boldsymbol{\theta}$ are equivalent to the abundances in that patient's gut microbiome:

$$\Pr\left(\hat{\boldsymbol{x}} = (\hat{x}_1, ..., \hat{x}_K)\right) = \mathbf{Multi}(\hat{\boldsymbol{x}} | M, \boldsymbol{\theta}) = \begin{cases} \dfrac{M}{\hat{x}_1!, ..., \hat{x}_K!} \prod\limits_{k=1}^{k} \theta_k^{\hat{x}_k} & \text{if } \sum\limits_{k=1}^{K} \hat{x}_k = M \\ 0 & \text{else} \end{cases}$$

The vector $\hat{\boldsymbol{x}}$ of species counts is transformed into a valid compositional dataset of relative abundances by dividing it through the number of total bacteria M. In order to model the gut microbiome composition of a group of patients, we need a probability distribution over abundance profiles. A fitting choice is the Dirichlet distribution, as it defines a probability measure over a set of variables $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$, which fulfill exactly the requirements that we need, referring to the compositional properties of lying on the $K - 1$ simplex (see Section 2.2). The Dirichlet distribution is defined as:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \mathbf{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \quad \sum_{i=1}^{K} \theta_k = 1 \quad \theta_k \in [0, 1]\ \forall k \in \{1, ..., K\}$$

The parameter vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)$ can be viewed as a descriptor encompassing all the factors influencing the abundance profile of patients, such as environment, diet, genetics, and other factors. The two distributions can be combined into the Dirichlet-Multinomial distribution, which models the probability of a sequencing result of M bacteria from one patient, given the scenario descriptor $\boldsymbol{\alpha}$:

$$\Pr(\hat{\boldsymbol{x}}|M,\boldsymbol{\alpha}) = \mathbf{DM}(M,\boldsymbol{\alpha}) = \int_{\boldsymbol{\theta}} \mathbf{Multi}(\hat{\boldsymbol{x}}|M,\boldsymbol{\theta})\mathbf{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha})d\boldsymbol{\theta}$$

Using this distribution, we can generate valid compositional data. This is realized as a three-step process. We first sample $\boldsymbol{\theta}$ from the Dirichlet distribution defined by parameters $\boldsymbol{\alpha}$. Then, we sample species counts $\hat{\boldsymbol{x}}$ from the subsequent multinomial distribution with $\boldsymbol{\theta}$ as probabilities and finally normalize with M, which is the number of trials in the multinomial. Repeated N times, this generates a valid compositional dataset of N samples and K features. This is the process outlined by the second loop in Algorithm 2.

As a next step, we need a mechanism to construct the parameter vector $\boldsymbol{\alpha}$. The steps for this are outlined in the first loop of algorithm 2. For this, I chose to define one Beta distribution per feature, where each alpha is sampled from $\alpha_k \sim \mathrm{Beta}_k(\beta_k^1, \beta_k^2)$. The feature specific set of parameters β_k^1, β_k^2 are drawn from two uniform distributions $\beta_k^1 \sim \mathcal{U}_1(a_1, b_1),\ \beta_k^2 \sim \mathcal{U}_2(a_2, b_2)$ defined by the parameters a_1, a_2, b_1, b_2 given to the prior. The reasoning for this method is less biologically motivated and more chosen to yield the best results with the Dirichlet-Multinomial sampling process. This is illustrated in figure 4.1. Visualized here is a comparison of abundances in the original microbiome dataset (top left) and probabilities θ_k sampled from the Dirichlet distribution with parameter vector $\boldsymbol{\alpha}$ generated in the DM-prior (bottom left), drawn from a standard normal distribution (top right) and drawn from a uniform distribution (bottom right). As shown clearly, the method in the DM prior generates probabilities that follow a similar distribution as the abundances in the original data. Which is a large number of zero or close to zero values and then a steady decrease in frequency on a logarithmic scale.

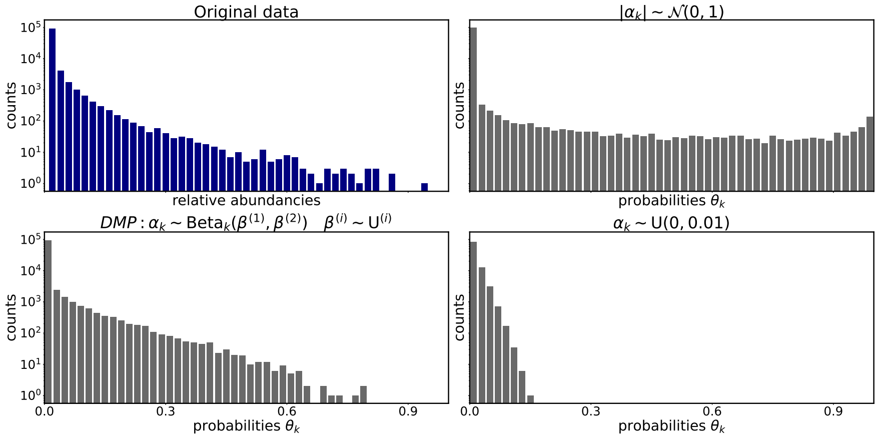


Figure 4.1: Histograms of: Abundancies in the original microbiome data (top left); Synthetic probabilities sampled from $\mathrm{Dir}(\boldsymbol{\alpha})$ with α_k generated according to DM-prior (bottom left); α_k drawn from $\mathcal{N}(0, 1)$ (top right); α_k drawn from $\mathcal{U}(0, 0.01)$

4.1.2 Mapping the data to labels

Mapping the data samples to labels is done via randomly instantiated bayesian neural networks (BNNs). This is based on the BNN prior proposed by [Hollmann et al., 2023]. The DM prior uses the same method to generate the labels, but the initial data is constructed using the procedure outlined in the previous section and some minor parameters are adjusted. This section explains the functionality of the label assignment in detail.

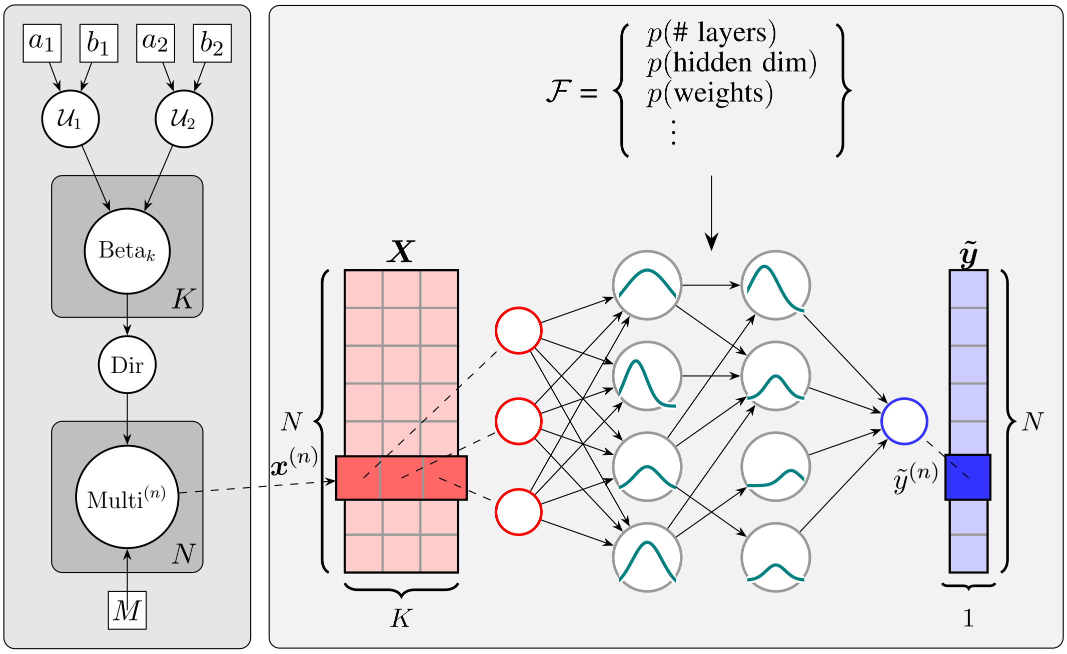


Figure 4.2: Dirichlet-Multinomial prior visualized. The data generation through the multi-step sampling procedure is shown left. One Bayesian Neural Network instance is drawn from the distribution over BNNs F. This network maps all data samples to a label. The BNN is sampled from a distribution over BNN instances including weights.

First, a BNN here refers to a Multi-Layer Perceptron with stochastic activations:

$$a_{i,j} \sim f_{\text{act}}(\mathcal{N}(\boldsymbol{w}_j^T \boldsymbol{a}_{i-1} + b_{i,j}, \sigma_{i,j})) \tag{4.1}$$

In contrast to a regular MLP, each activation is sampled from a normal distribution with a mean defined by its incoming connections, its bias, and a predetermined standard deviation. As a result, doing a forward pass with the same input multiple times generally leads to slightly different outputs. A BNN and its role in the DM prior is illustrated in figure 4.2. BNNs are meant to provide a complex and non-linear mapping from features of synthetic data samples to labels. One BNN instance $f_{BNN} \sim \mathcal{F}$ is used for one set of data X with N samples of K features. Every sample is passed into the network individually and assigned

a target by evaluating $\tilde{y}^{(n)} = f_{BNN}(\boldsymbol{x}^{(n)})$. While there is variation within the network from the stochastic activations and different inputs, the network weights do not change for the process of labeling one dataset. They are sampled once in the beginning and are not altered afterward. No training is performed, and the BNN is only used to perform forward passes. The BNN instance can be seen as one hypothesis that encapsulates some system of rules that relate features to a target variable. Thus, the weights need to be kept static so that the data are assigned labels using the same underlying mechanism. This is supposed to create classification problems with non-trivial, but learnable associations between features and a target variable. Furthermore, by having distributions of the various hyperparameters, the prior ensures that the relationship between features and label is sufficiently dissimilar between BNN instances and, therefore, datasets. This is necessary, as the PFN is intended to learn a general ability to perform inference, detached from any individual dataset.
The DM prior instantiates a BNN by drawing all necessary hyperparameters and network weights from predefined distributions. These can be seen in Table 4.1. The choice for the various distributions is a modified version of the SCM prior proposed by [Hollmann et al., 2023], with architecture hyperparameters being at the bottom of the table and more general procedural parameters at the top.

	Sampling distribution $p(\psi)$				
Pruning probability π	$0.9 \cdot$ Beta$(\beta^{(1)}, \beta^{(2)})$, where $\beta^{(1)}, \beta^{(2)} \sim$ Uniform$(0.1, 5.0)$				
		Max Mean $\hat{\mu}$	Min Mean $\check{\mu}$	$round$	min
#layers	TNLU	6	1	True	2
#hidden nodes/layer	TNLU	130	5	True	25
$\sigma_{i,j}$	TNLU	0.001	0.0001	False	0.0
σ_w	TNLU	10.0	0.01	False	0.0
		Choices			
Share $\sigma_{i,j}$ for nodes	Uniform Choice	{True, False}			
Act. functions	Uniform Choice	{Tanh, ReLU, Identity}			

Table 4.1: Hyperparameter distribution for instantiating BNNs. Table is taken and modified from [Hollmann et al., 2023]

First, the number of layers and nodes per hidden layer are drawn from a Truncated-Normal Log-Uniform distribution (TNLU). The TNLU, as defined by [Hollmann et al., 2023], samples a mean and standard deviation parameter from a Log-Normal $\mu, \sigma \sim \mathrm{LogUniform}(\check{\mu}, \hat{\mu})$. These are then used to sample a value from a Truncated-Normal distribution $v \sim \mathrm{TruncNormal}(\mu, \sigma^2, a = 0, b = \mathrm{inf})$ to ensure the value is non-negative. Next, weights are assigned to the network. This is done by sampling them from a normal distribution where the standard deviation is drawn from a TNLU:

$$w_{i,j} \sim \mathcal{N}(0, \sigma_w), \quad \sigma_w \sim \mathrm{TNLU}(\check{\mu}, \hat{\mu}, a, b) \tag{4.2}$$

Note that the weight standard deviation σ_w is only sampled once and then used for the entire BNN instance. Next, the activations are defined. This is done by picking an activation function as a uniform choice and assigning the standard deviation $\sigma_{i,j}$ for the stochastic activation to the nodes. $\sigma_{i,j}$ can either be identical for the whole network or sampled for each node individually. This is also decided by uniform choice. Lastly, the network is pruned to a sparsely connected BNN. For this, a pruning probability π is drawn from a Beta distribution, where the parameters $\beta^{(1)}, \beta^{(1)}$ are sampled from a uniform distribution as shown in the table. Then, all connections are either kept or set to zero using a binomial distribution with probability π. This pruning is restricted to connections between hidden layers only.
The labels generated by the DM prior are continuous, and for a classification, they need to be discretized. For this, we consider an entire set of N labels $\tilde{\boldsymbol{y}}$ that we divide into bins. As the DM prior is specially designed for binary classification, this is achieved by selecting a splitting point s and sorting the labels into the two resulting bins:

$$y^{(n)} = \begin{cases} 0 & \tilde{y}^{(n)} \leq s \\ 1 & \tilde{y}^{(n)} > s \end{cases} \tag{4.3}$$

The selection of splitting point s needs to be considered carefully as it determines the dataset imbalance. The selection of a splitting point is part of the curriculum training regime and is detailed in the next section.

4.2 Training regime

The splitting point that discretizes the continuous labels into two classes varies over the course of the training. It is drawn from a normal distribution, where the mean is the median of all class labels of one dataset, and the standard deviation increases over time. This is done by taking the standard deviation of the continuous class labels and scaling it based on the current epoch:

$$s \sim \mathcal{N}(\text{med}(\tilde{\boldsymbol{y}}), \epsilon \cdot \sigma_{\tilde{y}}) \text{ with } \epsilon = \left(\frac{\text{epoch}}{\text{total epochs}}\right)^2 \quad \text{and} \quad \sigma_{\tilde{y}} = \sqrt{\sum_{i=1}^{N} (\tilde{y}_i - \mu_{\tilde{y}})^2}$$

As a result, the splitting point is enforced to be close to the median in the beginning, and only as training progresses can the splitting point be sampled to fall further to either side of the median. This is illustrated in Figure 4.3. Datasets that are split exactly at the median are fully balanced, that is, they have approximately the same number of samples in each class. With increasing epochs, the standard deviation increases, and imbalanced datasets are more likely to be created. This process can be seen as a form of curriculum learning. The more imbalanced a dataset is, the higher the performance of an uninformed classifier that exclusively predicts the majority class, as seen in the training data. Consequently, beating the majority class predictor is more difficult as imbalance increases. Thus, the idea behind the proposed curriculum method is to first encourage the PFN to learn to predict labels based on the data without being able to default to the behavior of a majority class predictor. As being able to solve imbalanced learning tasks sufficiently is the desired behavior for the finished model, increasingly imbalanced datasets are introduced as the training progresses.

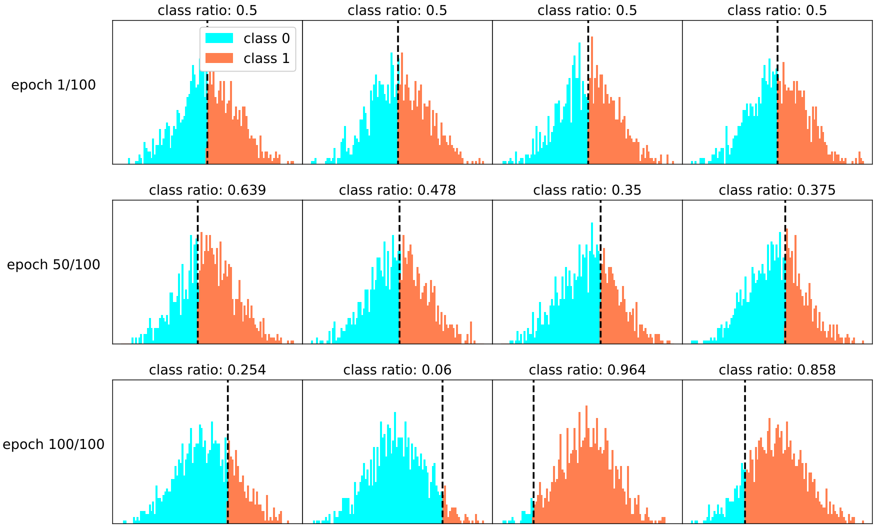


Figure 4.3: Showcase of the label discretization over the course of the training. Each histogram contains the continuous class labels of one synthetic dataset, with colors indicating the split into two discrete classes. Rows depict sample datasets from different stages during the training. Datasets are split evenly in the beginning, and with increasing epochs, the datasets are more likely to be imbalanced

.

In addition to the training curriculum of increasingly imbalanced datasets, cost-sensitive learning (see Section 2.1) is applied. The loss function is modified by weighting it according to the class distribution, where the weight is increased for the less frequent class and vice versa. The loss is now the weighted cross entropy loss:

$$l(\hat{\boldsymbol{y}}, y) = -w_y \log \frac{\exp(\hat{y}_{i=y})}{\exp(\hat{y_1}) + \exp(\hat{y_2})} \quad \text{with} \quad w_i = 1.5 - \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(y_n = i) \tag{4.4}$$

Here, $\hat{\boldsymbol{y}}$ is the vector containing the predicted probabilities for both classes, y is the correct class label, and w_y is the weight for the correct class. The weight for class i is calculated using its frequency in the dataset. As a result, the less frequent a class is, the higher it is weighted in the loss. The intention behind this is similar to the motivation for the curriculum learning, which is to improve the model's performance for datasets with highly imbalanced class distributions.

4.3 Interpretable machine learning for compositional data

The IML methods presented in Section 2.4 need to be adapted for compositional data. The summation requirement $\sum\limits_{k=1}^{K} \theta_k = 1$ causes features to be correlated, and simply removing a feature does not eliminate its information from the dataset. Concretely, since the feature values have to sum up to one, the value of feature k is encoded in the difference between the sum of the remaining features and one:

$$x_k = 1 - \sum_{j \in \{1:K\} \setminus k} x_j \tag{4.5}$$

To alleviate this, the datapoints are renormalized by their sum after removal of a feature. This leads to a modified calculation of the LOCO values from Equation 2.11:

$$\text{LOCO}_k = S_{ML}\left(\{\frac{\boldsymbol{x}_{K\setminus k}^{(n)}}{\sum_{j \in \{K \setminus k\}} x_j^{(n)}}\}_{n=1}^{N}, \boldsymbol{y}\right) - S_{ML}(\{\boldsymbol{x}_{1:K}^{(n)}\}_{n=1}^{N}, \boldsymbol{y}) \tag{4.6}$$

The same corrective measure is applied to the calculation of the ICE curves:

$$\phi_{k,t}^{(m)} = P_{ML}\left(\frac{\tilde{\boldsymbol{x}}_t^{(m)}}{\sum_{j=1}^{K} \tilde{x}_{j,t}^{(m)}}\right) \quad \text{with} \quad \tilde{x}_{j,t}^{(m)} = \begin{cases} \tilde{k}_t & \text{if } j = k \\ x_j^{(m)} & \text{else} \end{cases} \tag{4.7}$$

To explore the connections between features, I extend ICE curves to include two variables. For this, the feature space is expanded to $\kappa \times \kappa$. ICE curves are now defined as:

$$\phi_{k_1,k_2,t_1,t_2}^{(m)} = P_{PFN}\left(\frac{\tilde{\boldsymbol{x}}_{t_1,t_2}^{(m)}}{\sum_{j=1}^{K} \tilde{x}_{j,t}^{(m)}}\right) \quad \text{with} \quad \tilde{x}_{j,t_1,t_2}^{(m)} = \begin{cases} \tilde{k}_{t_1} & \text{if } j = k_1 \\ \tilde{k}_{t_2} & \text{if } j = k_2 \\ x_j^{(m)} & \text{else} \end{cases} \tag{4.8}$$

To generate the ICE plot, $\phi_{k_1,k_2,t_1,t_2}^{(m)}$ is evaluated on values of a feature grid. In theory, the two features k_1 and k_2 can have separate feature spaces, but for simplicity, all perturbed features follow the same space κ. The range of the feature space κ for both the one-dimensional and two-dimension ICE curves is chosen as $[10^{-7}, 10^{-1}]$. Importantly, the discrete feature space follows a logarithmic scale, as the abundances are also distributed logarithmically.

5 Experimental setup

This chapter provides details about the clinical microbiome dataset, evaluation procedure, hardware and architecture specifications and the considered baselines.

5.1 Microbiome dataset

The methods proposed in this thesis are evaluated on a large clinical gut microbiome dataset, with the presence of colorectal cancer (CRC) being the binary target variable. This dataset was sourced using whole genome sequencing [Ranjan et al., 2016] on stool samples of patients. This procedure identifies individual bacteria within a patients sample by their genomic profile and sorts them into species. The data is normalized to be compositional as described in Section 2.2. In the dataset one sample represents one patient and the label is assigned based on the presence of CRC. The result is a tabular dataset of compositional microbiome data with a binary class label. Overall the dataset consists of 11,462 samples with 1,391 features. A sample represents one patient and a feature corresponds to a known bacteria species that can be found in the human digestive system. Formally, the dataset can be described as:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\} \qquad N = 11,462$$

$$\mathbf{x}_i \in R_+^M,\ y_i \in \{0,1\},\ \sum_{j=1}^{M} x_{i,j} = 1\ \forall\ i \in (1, ..., N) \quad \text{with} \quad M = 1,391$$

Evidently, the dataset is highly imbalanced, as there are 10,761 healthy and 701 diseased samples present in the dataset. This results in an IR $= 0.939$. The dataset is further explored and visualized in Section 6.1.

5.2 Preprocessesing

Preprocessing was conducted for each fold individually. For this, all features with only one value were removed. The dataset contains no samples that can be considered obvious outliers therefore none were removed. To reduce the number of features to the maximum of 100 that TabPFN, TabForestPFN, and MedPFN are constructed to handle, I applied a one-way analysis of variance test [Ross et al., 2017] between each feature and the target variable. Features were then selected by their score. This analysis was performed on only the training data to prevent information from the test labels from leaking into the feature selection and, thus, evaluation procedure. Afterwards, the reduced samples are renormalized by the sum of the 100 kept features to retain compositionality.

5.3 Performance estimation

All methods and models in this thesis were tested using stratified K-fold cross-validation. For this, the data it randomly split into K folds of the same size while ensuring that each fold has approximately the same class distribution. Due to the high imbalance, the stratified split is necessary to prevent folds from having a large difference in the number of minority class samples. A model is trained on K-1 folds, and one fold is left out for testing. This is repeated K times and the results are averaged over all runs. To ensure comparisons of models and methods are unbiased, seeds were used for identical random split in all experiments. The number of folds K is set to 10. PFNs are trained with synthetic datasets of a particular size. While the architecture is flexible in the number of input samples and [Hollmann et al., 2023] showed TabPFN can extrapolate to longer inputs, the behavior of PFNs on larger datasets is potentially unstable. Thus, I limit the size of the cross-validation folds, such that the maximum input length to the size of the synthetic training sets seen by MedPFN during the pretraining is, which is 1024. To ensure the tests still encompass all samples, I repeat the above-described cross-validation experiment for all segments of the dataset until it is completely processed. This results in 10 roughly equal-sized cross-validation runs where stratification is again used to ensure equal distribution of test samples. Results are averaged over each segment.

As outlined in Section 2.1, several metrics should be examined for the evaluation of model performance on highly imbalanced datasets. The considered metrics are the accuracy, ROC AUC, and F_1-score. While the F_1-score is the harmonic mean of the precision and recall, both are also included at times for clarity. Additionally, the runtime for each experiment is listed. All measures are provided as means over the several individual experiments that make up the iterative cross-validation. The standard deviations are also denoted.

5.4 PFN architecture and training

The architecture of MedPFN is based on [Hollmann et al., 2023] and is explained in detail in Section 2.3.2. The model presented in this thesis has 12 encoder layers with 4 heads in each attention block and an embedding dimension of 256. The hidden dimension within the feed-forward networks of the transformer is 512. This results in a total of 6.48M parameters. The MedPFN model is trained on datasets with 1152 samples and 100 features from the DM prior. This dataset is split into a trainset of size 1024 and a test set of 128. The whole dataset is fed into the PFN and optimized with regards to the outputs for the test set, as described in 2.3.2. This is repeated for a total of 160.000 synthetic datasets, split into 50 epochs and with a batch size of 64.

The learning rate was initially set to 0.001 and follows a cosine learning rate schedule as proposed by [Hollmann et al., 2023]:

$$lr = 0.001 \cdot [\frac{1}{2}(1 + \cos(\pi \cdot \frac{epoch}{maxepochs}))]$$

This scheduling decreases the learning rate from 0.001 to 0 following the rate of decrease of a cosine wave between 0 and π. I chose AdamW [Loshchilov and Hutter, 2017] for an optimizer, based on its usage in the training of TabPFN.

5.5 Implementation and hardware

MedPFN was implemented using the TabPFN code as a baseline. All modifications to the TabPFN code or all other elements that are part of this research, including the DM prior, curriculum training regime, IML analysis, evaluation procedure, and data visualization, were implemented by myself. The pretraining of MedPFN was conducted on a single NVIDIA T4 GPU for a total of 6 hours. All evaluation experiments were run on an Intel i5-8600K CPU.

5.6 Baseline models

Several models were chosen to compare classification performance on the clinical dataset, including CatBoost [Prokhorenkova et al., 2018], XGBoost [Chen and Guestrin, 2016], random forest [Breiman, 2001], logistic regression [Kleinbaum et al., 2002], TabPFN [Hollmann et al., 2023] and TabForestPFN [den Breejen et al., 2024]. Also included is a majority class predictor to emphasize the difficulties of imbalanced classification problems. XGBoost's and CatBoost's hyperparameters were optimized for ROC AUC using grid search. The parameter grids are listed in Table 5.1. For random forest and logistic regression, the default parameters in the scikit-learn [Pedregosa et al., 2011] implementation were used. For both TabPFN and TabForestPFN I used the models provided by the authors in `https://github.com/automl/TabPFN` and `https://github.com/FelixdenBreejen/TabForestPFN` with default hyperparameters.

XGBoost		**CatBoost**	
Parameter	**Values**	**Parameter**	**Values**
Learning rate	[0.01, 0.1, 1.0]	Learning rate	[0.01, 0.1, 0.3]
Max depth	[5, 7, 9]	Depth	[6, 9]
Subsample	[0.5, 0.7]	Iterations	[100, 200]
N estimators	[5, 25, 50]	L2 leaf reg.	[1, 10]
Gamma	[0.1, 0.5, 0.9]		

Table 5.1: Hyperparameter grid for optimization of XGBoost and CatBoost

6 Experimental Results and Analysis

This chapter presents the results of all experiments conducted in this thesis. First, the microbiome data is analyzed in more detail and the ability of the DM prior to generate synthetic bio-medical datasets is visually inspected. Then, the performance of MedPFN on the microbiome dataset is tested extensively. Lastly, the decision-making process of MedPFN is explored through several IML procedures.

6.1 Dataset analysis

As a first step to better understand the classification task, it is advantageous to explore the microbiome dataset in more detail. Here, I will present details of the dataset and illustrate important characteristics. The dataset is depicted in figure 6.1. Displayed are the feature values for 500 healthy and 500 diseased samples chosen at random. The color of each pixel indicates the value at that position. Zero-values are shown in white, and values greater than zero are colored using a logarithmic coloring scheme. Multiple apparent attributes of the data can be seen in this image. Firstly, the high amount of white spaces indicates that

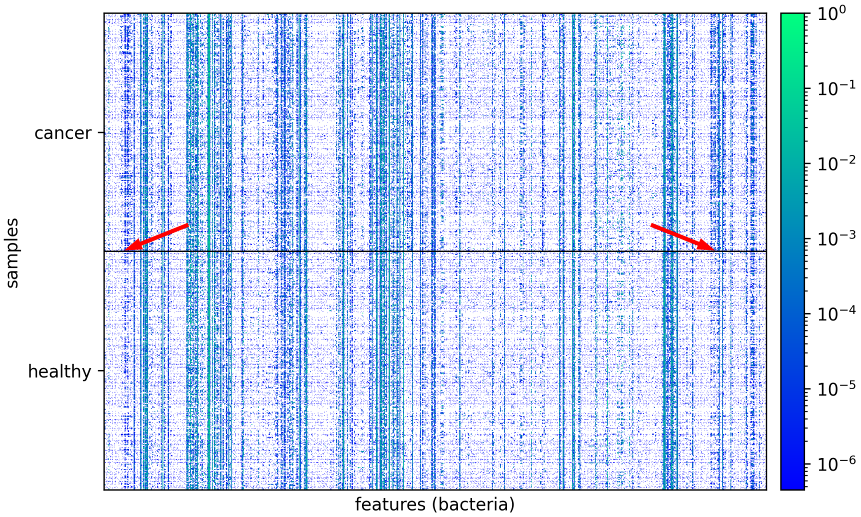


Figure 6.1: Feature values for 500 healthy and 500 diseased patients from the dataset. The pixel colors are determined by the respective values of the tabular dataset at that position with a logarithmic coloring scheme. Zero entries are white.

the dataset is very sparse. To further demonstrate this aspect, figure 6.2 depicts a histogram over the fraction of non-zero features per sample. As it shows, typically only around 10% of the considered bacteria species are present in a sample. This also entail, that patients not only differ in the frequencies of bacteria in their gut microbiome but also in the species that are present in general. Furthermore, in figure 6.1 distinct vertical lines are visible, and to a lesser extent, horizontal lines are discernible as well. The apparent and mostly continuous vertical lines indicate that some bacteria are present in high abundances in all samples for both patient groups.

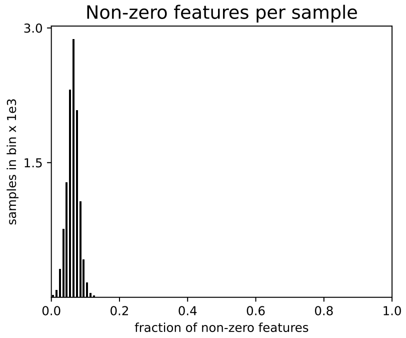


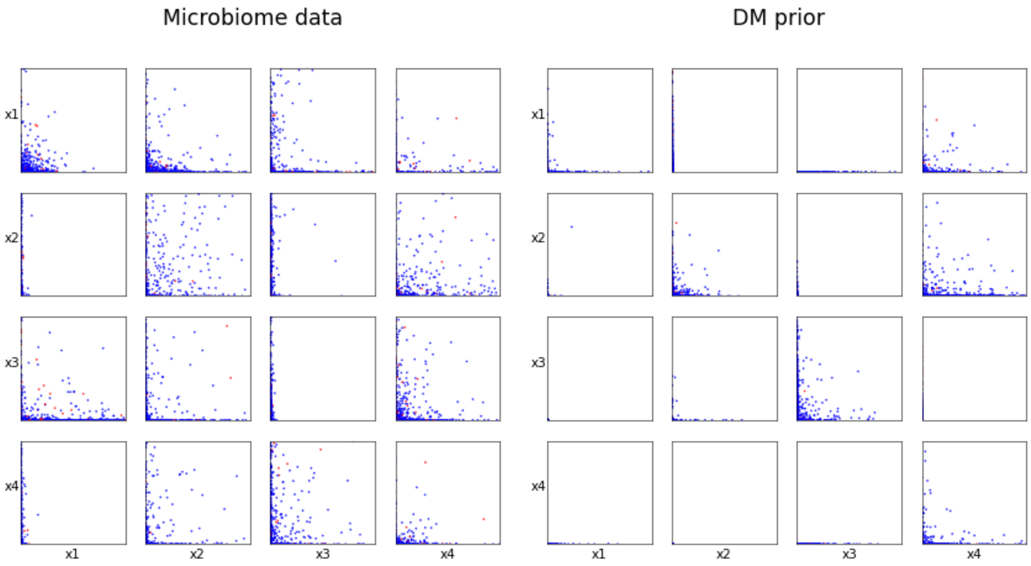
Figure 6.2: Plot showing the sparsity of the microbiome dataset. Only around 10% of features are non-zero for most samples.

The faint horizontal lines are formed by white spaces which implies, that some patients have a low diversity of bacteria in their gut. Lastly, the plot shows some vertical lines that are more pronounced in the diseased cohort than in the healthy individuals. The positions of these lines are indicated by the two red arrows.

6.1.1 Dirichlet-Multinomial prior

The DM prior was constructed to generate datasets that mimic compositional bio-medical data, like the microbiome dataset. Compositionality, as defined by Equation 2.7, is assured through the data generation process outlined in Section 4.1. Here, I present two visualizations that suggest that the DM prior generates datasets that share some similarities with the microbiome dataset.

Figure 6.3 shows grids that plot features from a dataset against one another. For each plot in a grid, two random features are selected, 1.000 samples are reduced to those two features, and then visualized as shown. Figure 6.3a is data from the microbiome dataset, and Figure 6.3b is data sampled from the DM prior.

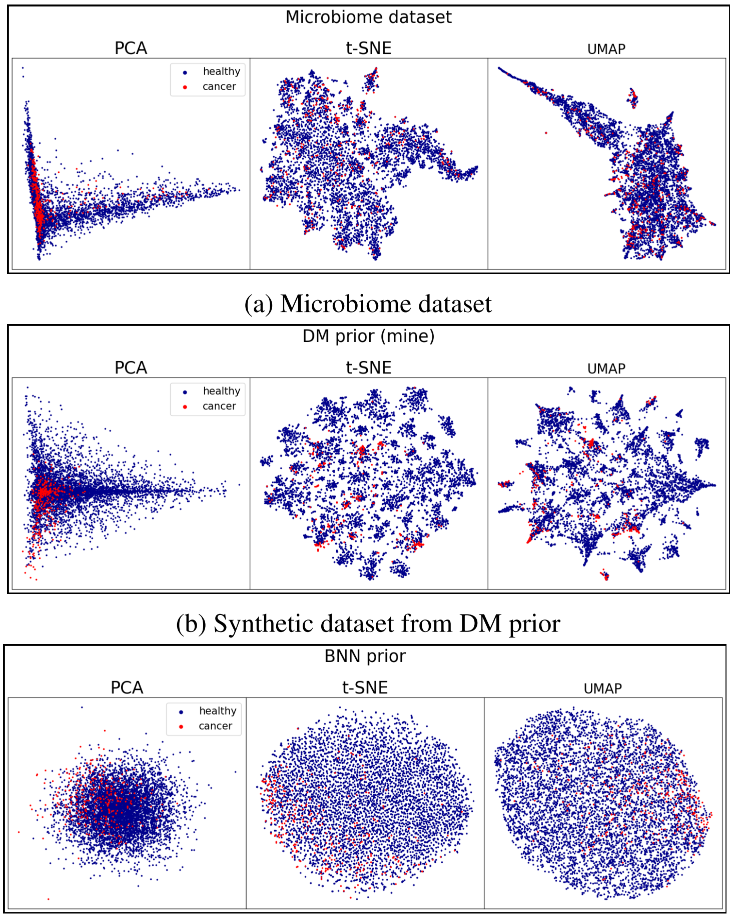


(a) Features of original microbiome dataset (b) Features of synthetic dataset from DM prior

Figure 6.3: Plot showing the relationship between random features from the Microbiome dataset (left) and a synthetic dataset sampled from the DM prior (right).

This plot shows that the synthetic datasets generated by the DM prior share characteristics with the microbiome data. There are many very sparse features where all or most values are 0. This can be seen in the plots where the points lie either projected onto one axis or are not visible at all. Feature combinations with more non-zero values are concentrated around the origin, with minor dispersion to either axes. A difference between the different data can be identified in the amount of dispersion that features exhibit. While this indicates there are some dissimilarities between the two, it is not required for the data to be identical. To further investigate the DM prior's ability to generate synthetic datasets imitating real biomedical data, I applied three dimensionality reduction techniques to entire datasets to visualize them in two dimensions. For this, I chose PCA [Maćkiewicz and Ratajczak, 1993], t-SNE [Van der Maaten and Hinton, 2008], and UMAP [McInnes et al., 2018]. The entire dataset was projected to two dimensions using the respective approach, and the samples colored based on their class label. The results can be seen in Figure 6.4. Depicted here are the original dataset (Figure 6.4a), a synthetic dataset from my DM prior (Figure 6.4b), and a synthetic dataset from the BNN prior ((Figure 6.4c)). The BNN prior from [Hollmann et al., 2023] utilizes the same label assignment process, but the features are sampled from a normal distribution instead of the Dirichlet-Multinomial.

This visualization clearly depicts the similarity between data generated by the DM prior and the microbiome dataset, especially when compared to the BNN prior. The PCA of the former two matches very closely, with two narrow shapes in the approximate direction of the principal components. In contrast, the data from the BNN prior is a circular shape in two dimensions. The 2D projection using t-SNE and UMAP shows a little more divergence



(a) Microbiome dataset

(b) Synthetic dataset from DM prior

(c) Synthetic dataset from BNN prior

Figure 6.4: Datasets projected into two dimensions using PCA, t-SNE, and UMAP. This is done for the original microbiome data (top), synthetic data from the DM prior (middle), and synthetic data from the BNN prior (bottom).

between the original dataset and the synthetic data of the DM prior. However, different clusters can be identified in both and as with the PCA, the difference is very minor when compared to the BNN prior. Notably, the classes seem to be less dispersed in the t-SNE and UMAP projections of the DM prior data when compared to the microbiome data. This indicates that the label-generation process might not be ideal. However, overall, this visualization suggests that data generated by the DM prior is reasonably similar to a real bio-medical dataset and a strong improvement from the BNN prior in that regard.

6.2 MedPFN performance comparison

In this section, I will explore the performance of my methodology on the microbiome dataset compared to several baselines. Figure 6.5 illustrates the results of a 10-fold cross-validation test of all models on the full microbiome dataset. Table 6.1 lists the exact values of the conducted experiments. MedPFN is the regular model with 10 internal ensembles, MedPFN-small does not use ensembling, and MedPFN-FT additionally includes finetuning, as laid out in Section 2.3.3.

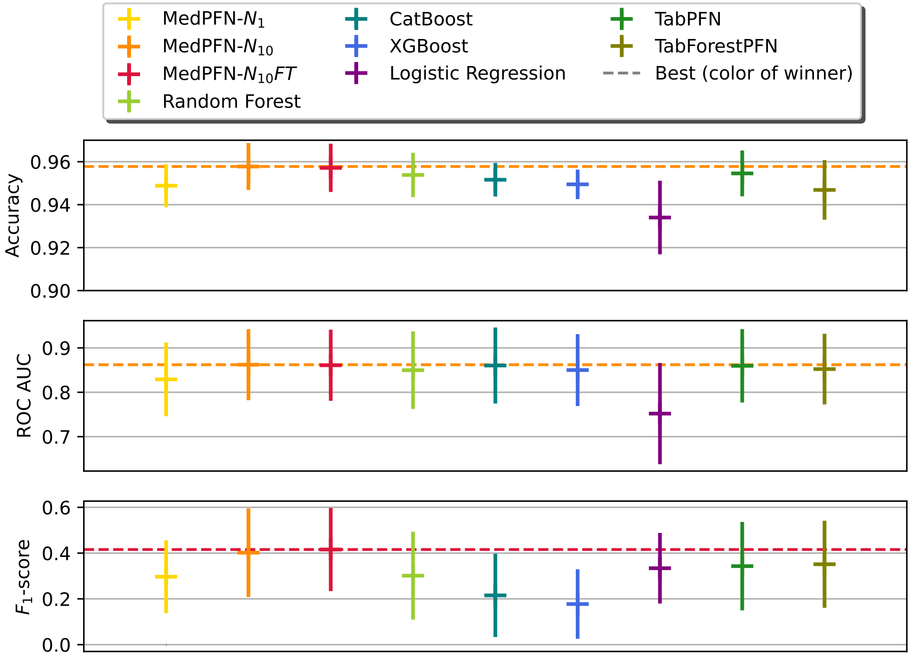


Figure 6.5: Comparison between MedPFN and baseline models on accuracy, ROC AUC and F_1-score. Center position indicates mean over cross-validation and vertical line length encapsulates standard deviation. Horizontal dashed lines show score and identity of winner.

MedPFN seems to provide the most reliable predictions on the clinical microbiome dataset in this experiment. The regular model without finetuning, MedPFN, has the best mean accuracy of 0.958, the best precision of 0.748 and the best ROC AUC with 0.862. The model that additionally uses the finetuning procedure achieved the highest F_1-score with 0.416. However, the margin of the measured ROC is of MedPFN to models such as CatBoost or TabPFN is very minor. In contrast, the difference in F_1-score of considerable magnitude. My model without finetuning also scores substantially higher on F_1-score than the next best baseline. The only metric where my model does not score highest is recall. Here, logistic regression has a slightly better score, but it is uncompetitive in all other metrics. Random forest has a better performance than logistic regression but lacks in

ROC AUC and recall. CatBoost and XGBoost both demonstrate high ROC AUC, which they were optimized for using hyperparameter grid search, but they fail to come close to other models in precision and recall. The PFN-based models seem to perform best overall. With TabPFN and TabForest being fairly close to MedPFN-N_{10} is all metrics. Interestingly, TabForest shows a similar trend from TabPFN compared to my model with and without finetuning. The recall rises and consequently, the F_1-score slightly increases as well, but this deteriorates the performance in accuracy, precision, and ROC AUC. This implies that finetuning has a similar effect for the different models in this setting. MedPFN demonstrates to have a decent runtime. The slight increase from TabPFN is due to the higher number of ensembles. CatBoost and XGBoost both incur a higher runtime than my model, although this includes the hyperparameter search. Random forest is exceedingly fast and also demonstrates a decent performance. The shortest runtime is shown by logistic regression, but evidently, it can not compete in predictive quality with the best models.

Model	Accuracy		Precision		Recall		ROC AUC		F_1-score		Runtime	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ sec.	σ sec.
Majority class	0.946	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000
Log. regression	0.934	0.017	0.386	0.191	**0.313**	0.151	0.752	0.114	0.334	0.154	**0.007**	0.001
Random forest	0.954	0.010	0.655	0.381	0.207	0.142	0.85	0.087	0.301	0.192	0.194	0.007
CatBoost	0.952	0.008	0.449	0.361	0.150	0.132	0.860	0.086	0.215	0.182	26.477	2.111
XGBoost	0.949	0.007	0.338	0.264	0.127	0.115	0.850	0.081	0.177	0.152	5.838	0.251
TabPFN	0.955	0.011	0.655	0.319	0.245	0.150	0.860	0.083	0.343	0.193	2.586	0.045
TabForestPFN	0.947	0.014	0.522	0.281	0.288	0.180	0.852	0.080	0.351	0.190	17.392	0.491
MedPFN-small	0.949	0.010	0.550	0.308	0.222	0.130	0.829	0.083	0.297	0.159	0.278	0.031
MedPFN-FT	0.957	0.011	0.742	0.292	0.310	0.161	0.861	0.080	**0.416**	0.182	22.725	1.685
MedPFN	**0.958**	0.011	**0.748**	0.297	0.293	0.164	**0.862**	0.080	0.402	0.194	3.058	0.214

Table 6.1: Table displaying the results of a full cross-validation run on the microbiome dataset using various models. Each metric is given with a mean and standard deviation. MedPFN-small does not include ensembling, and MedPFN-FT adds finetuning procedure.

This experiment demonstrates that ensembling improves the quality of the predictions of the MedPFN model, as MedPFN beats MedPFN-small in all metrics. In fact, without ensembling the performance of MedPFN falls behind most of the baselines. Finetuning slightly improves the F_1-score, but reduces accuracy and ROC AUC. This seems to stem from a higher recall, but diminished precision. Both ensembling and finetuning entail a runtime cost. The ensembling seems to increase the runtime by the factor of used ensemble models. Finetuning for 40 steps increases the training time substantially as well.

6.2.1 Isolating method effects

To understand to what extent the proposed Dirichlet-Multinomial prior and training regime affect the results, I trained two modified instances of the model. MedPFN-Balanced does not utilize the curriculum training regime described in 4.2. Instead, the datasets generated using the DM prior were discretized into two equally sized bins, creating only fully balanced classification tasks. This was kept static during the entire pretraining procedure. The second instance, MedPFN-BNN, used the training regime, but the datasets were generated using the regular BNN prior. Meaning that the features were sampled from a standard normal distribution and not the Dirichlet-Multinomial that mimics compositional bio-medical data. All other parameters of the model, the prior and the training regime are identical to the regular model. Table 6.2 lists the result of both variations and the full model on the microbiome dataset. As the finished models are identical in architecture, the runtimes are excluded.

Model	Accuracy		Precision		Recall		ROC AUC		F_1-score	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
MedPFN-Balanced	0.897	0.028	0.296	0.098	**0.622**	0.198	0.860	0.084	0.396	0.124
MedPFN-BNN	0.956	0.009	0.718	0.327	0.230	0.141	0.839	0.084	0.334	0.181
MedPFN	**0.958**	0.011	**0.748**	0.297	0.293	0.164	**0.862**	0.080	**0.402**	0.194

Table 6.2: Models trained without parts of the proposed method compared to the full model. MedPFN-Balanced was trained only with fully balanced synthetic datasets. MedPFN-BNN is trained with BNN prior instead of DM. MedPFN is the regular model. All other parameters are kept the same.

The results suggest that both methods have a positive effect on model performance. Leaving either the DM prior or the curriculum training regime out degrades the quality of the predictions. The full model achieves the best mean scores on accuracy, precision, ROC AUC, and F_1-score. The only outlier in this trend is the recall, where MedPFN-Balanced has by far the highest. But conversely, MedPFN-Balanced displays very low accuracy and precision. These scores indicate that it predicts a much higher number of positive classes than the other models. As this model was pretrained on fully balanced datasets only, it is intuitive that the model is biased towards predicting similar amounts of both labels. The model with a BNN dataset prior substituted for the DM prior demonstrates a fairly similar pattern in its scores to the full model. However, every metric is slightly worse than the full model. While this difference is not as substantial as when dropping the training regime, it still implies that the DM prior increases the performance on this particular dataset.

6.2.2 Imbalance analysis

MedPFN is specially developed for highly imbalanced datasets. The training regime is designed to improve performance on datasets where one group dominates the class distribution. In order to further understand the impact of the methodology, I performed the following analysis. The dataset was artificially rebalanced for different class distributions. This was done by randomly removing samples belonging to one class until the desired IR (see Equation 2.1) was attained. MedPFN, random forest, logistic regression, and TabPFN were evaluated on modified datasets with IR $\in [0.5, 0.99]$. Then, the mean over the baseline models was calculated for the Accuracy, ROC AUC and F_1-score. The means were then subtracted from the scores of MedPFN. The results are shown in Figure 6.6. This plot is meant to illustrate the change in relative performance between MedPFN and the

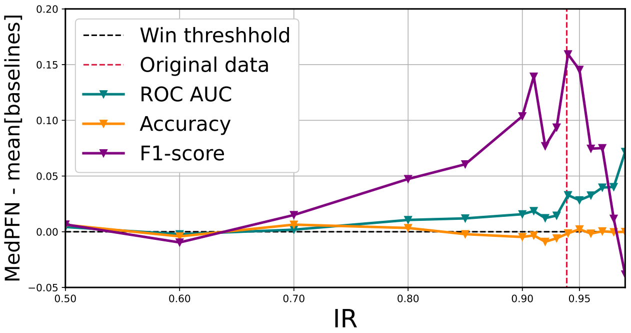


Figure 6.6: The plot shows the disparity between MedPFN's performance and those of baseline models logistic regression, random forest, and TabPFN on differently imbalanced datasets. The class distribution of the microbiome dataset is artificially adjusted and the models performance tested.

baselines with increasing IR. As shown, MedPFN's performance relative to the baselines rises as imbalance increases. This is true for all three metrics. The difference is most notable for the F_1-score, where the difference rises up to 0.15 for very imbalanced datasets. For IR> 0.95, this difference sharply drops, indicating that MedPFN's performance might be unstable there. Overall, this experiment demonstrates that MedPFN excels at the highly imbalanced datasets it was designed for but has diminishing performance compared to baseline models for more evenly balanced classification tasks. The behavior of the model for differently imbalanced datasets is further analyzed in Figure 6.7.

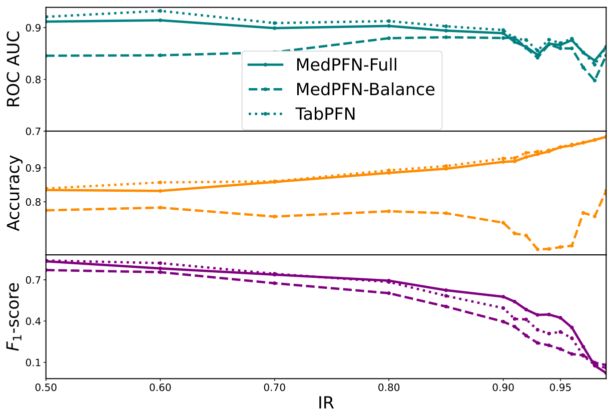
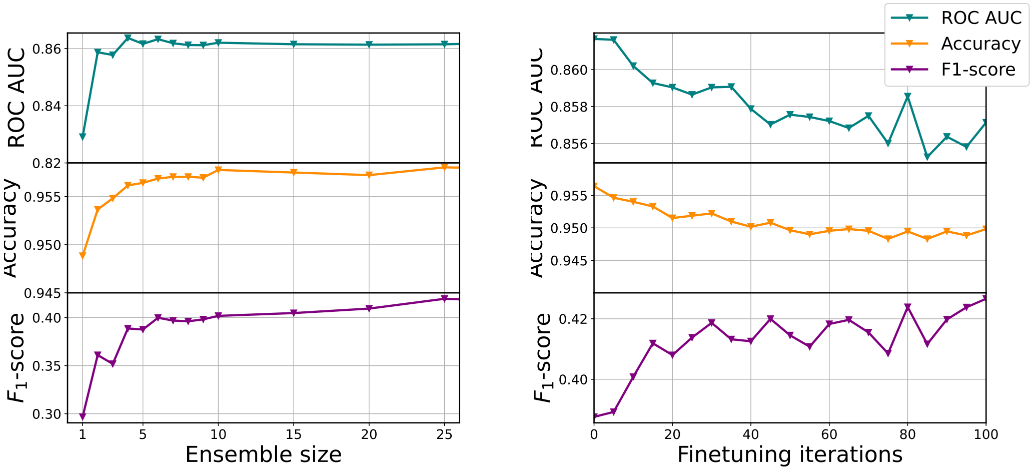


Figure 6.7: The plot shows MedPFN's specialization on imbalanced datasets. The class distribution of the microbiome dataset is artificially adjusted, and the models' performance tested. The plots are the difference of each metric between MedPFN and the mean of logistic regression, random forest, and TabPFN.

Illustrated here are the individual ROC AUC, accuracy, and F_1-score for MedPFN, MedPFN-Balanced, and TabPFN over the differently rebalanced datasets. The plots show that TabPFN outperforms MedPFN for most of the values of IR on all metrics. Only for very imbalanced datasets do accuracy and ROC AUC catch up. Mirroring the results shown in Figure 6.2, the most substantial improvement of MedPFN comes in the F_1-score, where it exceeds TabPFN for IR> 0.8. Surprisingly, MedPFN-Balanced, the model pretrained on only fully balanced datasets, performs consistently worse over the entire range of values for IR. This includes the fully balanced dataset, even though MedPFN-Balanced was trained on only fully balanced synthetic datasets. The implication is that the curriculum training regime not only enhances the model's ability to classify imbalanced datasets but improves the predictive performance overall.

6.2.3 Ensembling and finetuning

Ensembling within a PFN uses multiple perturbed copies of the data to elicit varied outputs from one model and then averages over them for a more robust prediction. Finetuning performs a small amount of optimization steps on the model using the training data. Both methods are intended to improve performance. Table 6.1 indicates that ensembling is crucial in MedPFN's strong performance, but finetuning is less valuable. In order to explore the effects of both in more detail, figure 6.8 illustrates the ROC AUC, Accuracy, and F_1-score for increasing numbers of ensembles and finetuning steps separately. Figure 6.8a displays the results of MedPFN with different numbers of ensembles ranging from 1 to 25. As shown, there is a sharp increase in the ROC AUC, accuracy, and F_1-score



(a) MedPFN scores with increasing ensemble size.

(b) MedPFN score with increasing finetune steps.

Figure 6.8: Plots showcasing the effect of different ensemble sizes and the number of finetuning steps.

when moving from 1 to 2 members in the ensemble. This increase continues, although less drastically, as the number of members rises. After 10 members, the F_1-score still improves slightly. However, accuracy and ROC AUC stay mostly flat. Overall, ensembling clearly has a strong positive effect on model performance, but the difference of increasing ensemble size rapidly declines after around 5.
Figure 6.8b lays out the effect of the finetuning procedure with up to 100 steps and exposes a clear trend. Finetuning improves the F_1-score, but simultaneously degrades the model's ROC AUC and accuracy. Consequently, finetuning provides mixed results in this setting. If the F_1-score is considered the most important performance metric, it might be a worthwhile tradeoff, but as finetuning also comes with additional runtime cost, it needs to be carefully considered.

6.3 Interpretable machine learning

This section investigates the decision-making process of MedPFN using interpretable machine learning methods.

6.3.1 Feature importance

Feature importance aims to measure the impact of features on the classification performance in a machine learning model. Here, I apply leave-one-covariate-out analysis to gauge the importance of individual features in MedPFN's predictions. The results are displayed in Figure 6.9.

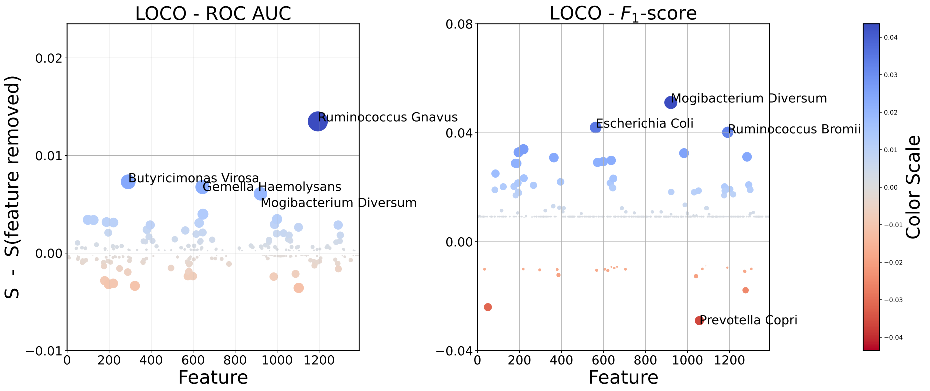


Figure 6.9: Leave-one-out (LOO) feature importance analysis. Each point represent the score of the model on the full dataset minus the score when removing one feature. Left for ROC AUC and right for F_1-score. Colors indicate the sign of change. Point opacity and size indicate the magnitude in change.

Shown here, are the differences between the model's performance on the full dataset and when removing an individual feature. On the left is the change in measured ROC AUC and on the right is the change in F_1-score. Each point corresponds to one feature, and its position in the y-direction expresses the decrease in performance when removing that feature. This change is emphasized with the size and opacity of the point for clarity. The color indicates the sign of the change.

Both plots display a similar pattern. Most importance values lie close to the center, and only a handful show a significant importance. The fact that so many features cause a change in predictive performance is not necessarily their direct influence but possibly a result of the renormalization done after removal of a feature. When a feature is removed, the other feature values have to be increased to retain compositionality. The distinct horizontal line in the F_1-score LOO plot is likely an artifact of this. Notably, some features also impose a negative effect, as the score with the feature removed is higher. This implies that some features cause a decrease in predictive quality of the model. In general, the magnitude of the changes is limited to 0.015 for the ROC AUC and 0.05 for the F_1-score. While those indicate a measurable impact on the model's performance, no feature can be identified as a dominant source of information for the prediction. This, in turn, hints that MedPFN utilizes combinations of several features in its decision-making process. Interestingly, features have dissimilar importance in the two metrics. For example, removal of *Ruminnococcus Gnavus* clearly has the greatest impact on the model's ROC AUC, but does not seem particularly impactful for the F_1-score.

Some of the most important species have been annotated, and many of them have been found to be associated with the presence of CRC in previous research. This includes *Ruminnococcus Gnavus* [Crost et al., 2023], *Mogibacterium Diversum* [Chen et al., 2012], *Escherichia Coli* [Wassenaar, 2018] and *Ruminococcus Bromii* [Ulger et al., 2024]. For *Prevotella Copri*, an association has also been reported [Ulger et al., 2024], but removing it increases the F_1-score. This implies, that the model at least identifies it as significant in the training data, even if it degrades the inference quality.

I conducted an additional experiment to support the notion that MedPFN's predictions are not overly dependent on individual species of bacteria. The results are illustrated in Figure 6.10.

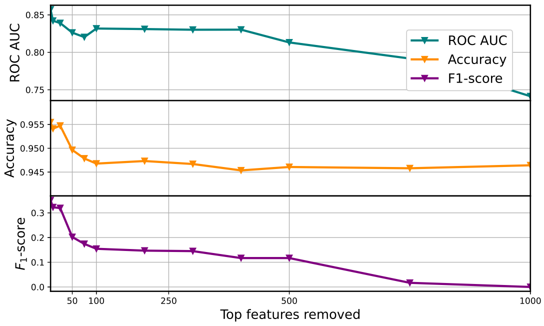


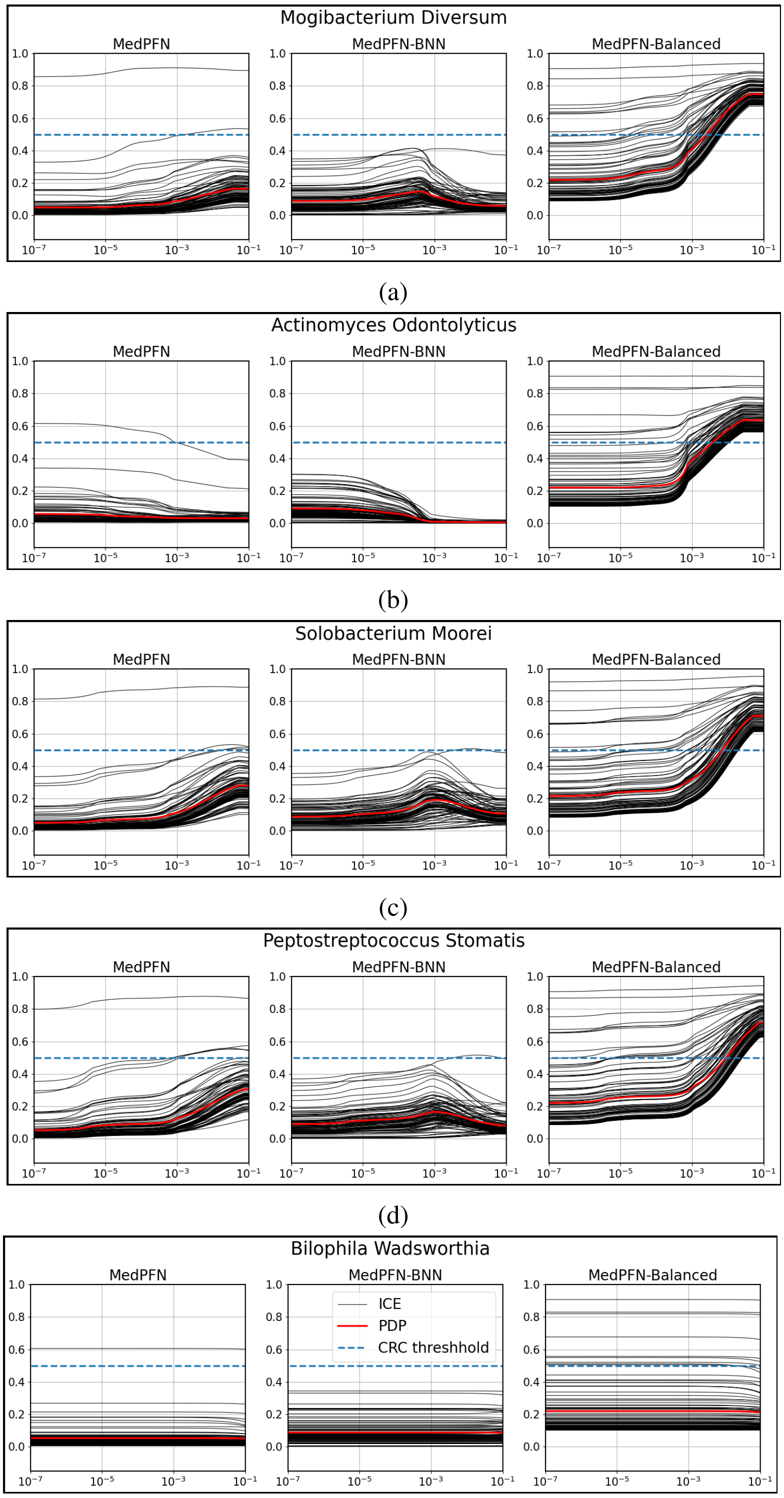
Figure 6.10: MedPFN performance on the microbiome dataset when removing features in order of highest importance as measured in LOCO analysis.

Features were iteratively removed in order of highest importance according to the LOCO analysis. The measured ROC AUC, accuracy, and F_1-score during this process. Removing features causes a clear decrease in model performance. The metrics drop consistently until 100 features have been removed. However, this drop is gradual and evidently does not hinge on just a single feature. The model scores then drop very slowly to 500 features removed. The ROC AUC and F_1-score only entirely deteriorate towards the removal of the top 1.000 features. Overall, this experiment supports the finding of the LOCO analysis that the model uses a diverse combination of features for it's predictions. However, it is possible that a feature removal based on a different ranking would lead to a more abrupt drop in performance.

6.3.2 Feature effect

By adjusting the values of a feature and measuring the predicted probability after each change, ICE curves can give more insights into the effect individual features have in a model's prediction. ICE curves for various bacteria species are displayed in Figure 6.11. Included here are the predicted probabilities for the three models from Section 6.2.1 in order to compare their decision-making process and identify causes for their differing performances. The five bacteria species shown here are *M. Diversum*, *A. Odontolyticus*, *S. Moorei*, *P. Stomatis*, *B. Wadsworthia*, and *F. Nucleatum*. Each black curve is the model's predicted probability for one patient, with the x-axis denoting the artificially induced feature value. The red curves are the PDP plots that average over all patients. The decision boundary is denoted by the blue line. The species were chosen because they demonstrate a significant effect on the model's predictions, with the exception being *B. Wadsworthia*. This species was included as a control example and showcases ICE curves for a feature that is evidently not used in the prediction.

All three models show distinct patterns for the influence the various features have on their decision-making process. MedPFN-Balanced predicts a much higher baseline probability of CRC for most patients. This can be seen in comparing the starting positions of the individual samples to those of the other two models. This implies, that MedPFN and MedPFN-BNN are more biased towards the majority class of no CRC. This suggests that the curriculum training regime works as intended. We can see a clear influence on the predictions of all models following the increase of the feature values. The strongest reaction can be observed for MedPFN-Balanced. For high values of the four significant species, all samples are given a probability greater than 0.5 and are thus classified as CRC. In contrast, the effect of the feature perturbation in the other two models is more restrained. Only a few samples are given high probabilities, even for very high feature values. The ICE curves for MedPFN and MedPFN-BNN also show different characteristics. While the four bacteria seem to have a monotonic influence on MedPFN, for MedPFN-BNN, the predictions first rise and then fall for 3 of the features. This illustrates a clear influence of the choice of dataset prior on the model's inference process, the direct cause of the observed behavior is not as easily concluded. The reason for this drop is not necessarily the increase of the considered feature but possibly the renormalization of the other features.



(a)

(b)

(c)

(d)

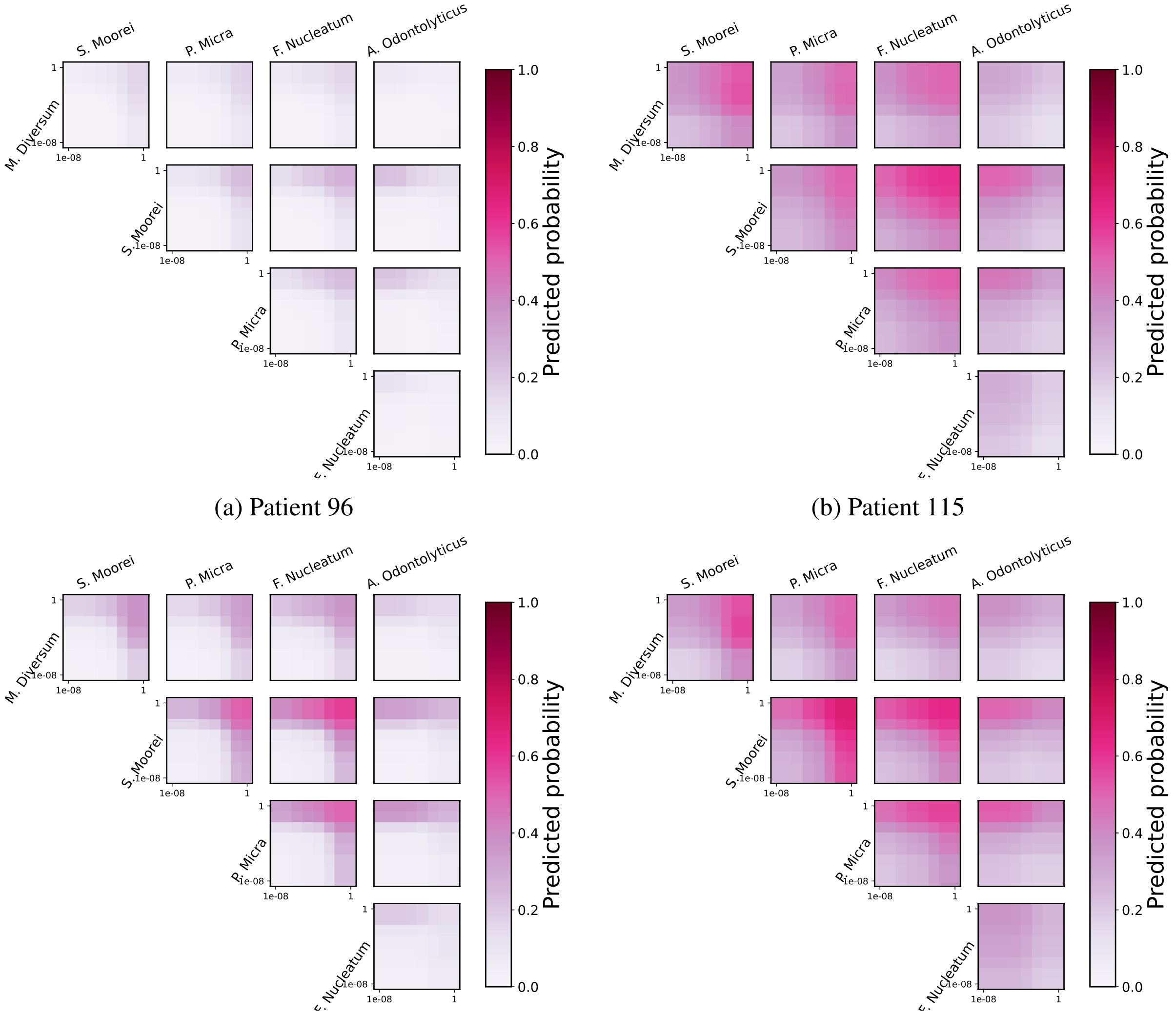
(e)

Figure 6.11: ICE curves (black) and PDPs (red) of MedPFN, MedPFN-BNN, MedPFN-Balanced for 5 different bacteria species. Blue line is decision boundary.

Furthermore, some samples in the ICE plots for MedPFN and MedPFN-BNN stand out by going against the direction of the majority of samples. This suggests that the influence of the displayed features in the decision-making process of these models depends other aspects of the sample. Conversely, MedPFN-Balanced demonstrates much more even ICE plots. This supports the possibility that the curriculum training encourages the model to learn more complex inference patterns. Comparing the different bacteria, *M. Diversum*, *S. Moorei*, *P. Stomatis*, and *F. Nucleatum* seem to be positively associated with CRC. *A. Odontolyticus* is negatively associated with CRC for MedPFN and MedPFN-BNN. In contrast, MedPFN-Balanced judges this species to be positively correlated. As for the LOCO analysis, the associations between these species and CRC is supported by previous research. Positive associations with CRC presence in a patient have been found for *M. Diversum* [Chen et al., 2012], *P. Stomatis*, [Osman et al., 2021] *S. Moorei* [Narii et al., 2024], and *A. Odontolyticus* [Narii et al., 2024]. The last finding contradicts the relationship depicted in Figure 6.11b, as both MedPFN and MedPFN-BNN show a negative association with *A. Odontolyticus*. Only the ICE plots of MedPFN-Balanced support this finding. However, it is not certain if this is the association also present in our data.

ICE curves can be extended to two features in order to explore the combined effects of features. The outcome of this analysis is illustrated in figure 6.12 using the regular MedPFN model. Five bacteria, M. Diversum, *S. Moorei*, *P. Micra*, *F. Nucleatum*, *A. Odontolyticus* were tested in a combinatory fashion, resulting in ten plots when removing duplicates and the same feature. The result is shown for four different patients, which were chosen to demonstrate the diverse outcomes of this intervention. The color scheme in all plots has the same range, such that pixels of the same color for different patients correspond to the same probability. The bottom left corner of each heatmap is the lowest value I set for both bacteria. The feature values then increase on a logarithmic scale when going up and to the right. Each subplot displays a distinct pattern. Figure 6.12a shows a patient where the model firmly predicts a very low probability. Increasing the feature value for any of the evaluated species leads to very little change, even in combination. In comparison, patient 119 in figure 6.12c similarly has a low initial probability, but the model's prediction increases substantially with higher values of *S. Moorei*, *P. Micra* and *F. Nucleatum*, especially when combined. The other two patients show a much higher baseline probability. However, the increase of certain bacteria values produces a stronger effect in patient 272 in figure 6.12d than in patient 115 in figure 6.12b. Overall, we can see that features have additive effects. Most notably, *S. Moorei*, *P. Micra* and *F. Nucleatum* show high probabilities for three patients when any of their combinations is increased simultaneously. The most significant pairing varies over patients. For patient 115 in 6.12b, the most relevant pair seems to be *S. Moorei* and *F. Nucleatum*. In contrast, for patient 272 in 6.12d, *S. Moorei* and *P. Micra* show the highest probability for their maximum value. A notable outlier is *A. Odontolyticus*, which has a negative effect on the model's predicted probability. This is the same pattern as observed in Figure 6.11b. Increasing the value for that species causes the probability to fall for all combinations with the other four bacteria in all patients.

Overall, the outcome of simultaneously increasing the abundances of two species seems to



(a) Patient 96

(b) Patient 115

(c) Patient 119

(d) Patient 272

Figure 6.12: 2D ICE plots of MedPFN for four different patients

be an additive effect of the patterns observed in the one-dimensional ICE of MedPFN in Figure 6.11. At least for the bacteria and patients shown here, there is no example of high presence of one bacterium inverting the effect of another or any relationship of that nature. Similar to the one-dimensional ICE curves, this experiment shows that the bacteria are of varying importance for different patients in the decision-making process of MedPFN. Increasing the abundance of a species the model assumes to be associated with CRC does not change the predictions for all patients, and the observed effect differs in strength between them. This indicates that the inference process takes complex relationships into account and is not based on entirely individual important features.

7 Discussion

The research conducted in this thesis indicates that the proposed DM prior and curriculum training regime improve the predictive performance of a PFN model highly imbalanced classification tasks with compositional, bio-medical data. The data generation process through the combined Dirichlet-Multinomial distribution creates data visually similar to the clinical microbiome dataset, as seen in Section 6.1. Whether the correlations between features and a target variable are comparable to that of real bio-medical data could not be ascertained. Section 6.2 provides empirical evidence that MedPFN surpasses baseline ML models, like logistic regression, random forests, gradient-boosted trees and other PFNs, on predicting the presence of CRC in a clinical gut microbiome dataset. The excellent performance is expressed in all considered metrics and comes with no runtime cost exceeding that of comparable baselines. The ensembling of MedPFN with copies of itself, as proposed by [Hollmann et al., 2023], strongly improves the performance. The finetuning procedure of [den Breejen et al., 2024] is not beneficial in this context. Section 6.2.1 explored the impact of the two main contributions, the DM prior and the curriculum training regime, on the model's performance. Two identical models, but one trained without the curriculum training regime and another trained without the DM prior, both fall short of MedPFN's capability on the considered dataset. The findings here suggest that both methods have a positive effect on the model's performance. As shown, the curriculum training regime is vital in ensuring the model is well calibrated for highly imbalanced classification tasks. The model without the training regime is overly biased towards balanced classification tasks. Substituting the DM prior for a BNN prior similarly negatively impacts the performance. This implies that the synthetic bio-medical datasets generated by the DM prior help to improve the proficiency of MedPFN this type of data. Section 6.2.2 demonstrated that MedPFN is indeed specialized for highly imbalanced datasets. The performance difference to other models increases with the imbalance of the dataset. Interestingly, the model MedPFN-Balanced trained without the curriculum scores below the regular MedPFN even on fully balanced datasets. This is despite the fact, that the MedPFN-Balanced was trained only on fully balanced synthetic datasets. This result suggests, that the training curriculum generally improves the models inference capabilities. A possible explanation is that imbalanced datasets present a more challenging classification task, and including those in the pretraining encourages the model to learn more complex inference strategies. These findings are supported by the results in Section 6.3.2. The ICE curves of MedPFN-Balanced show a clear bias towards balanced datasets and additionally hint at a less complex decision-making process, as the curves are very similar for all patients. The ICE curves also confirm a clear difference in the decision-making process of the model with and without the DM prior.

Section 6.3 explored how features influence the predictions of MedPFN using interpretable machine learning methods. Feature importance was measured using leave-one-covariate-out analysis and showed that while several features are of higher importance than others, the model seems to derive predictions by combining the information from multiple species of bacteria. This result and the slow decline in model performance when the most important features are removed suggest that MedPFN is likely to be robust against data with missing features or noise. This is a valuable property for a potential medical application, but the experiments have not shown this robustness conclusive or unique to MedPFN over other models.

ICE curves were used to gain more insight into details of how individual features affect the model's predicted probability CRC being present in a patient. The diversity in ICE curves for the different species of bacteria hints that MedPFN learns varying interactions between features and the target variable. As shown, different levels of abundance are deemed important for the considered species. The ICE curves also illustrate a visible influence that the DM prior and curriculum training regime have in the learning decision-making process. Models trained without the two methods show a clear difference in their response to feature perturbations. Lastly, two dimensional ICE plots illustrate a general additive influence of bacteria in the model's predictions. Also, species have divergent patterns of influence in the model's prediction for different patients. This supports the notion that even when a species of bacteria is identified to be associated with CRC, the model utilizes this information asymmetrically among patients. Distinguishing between patients when performing inference is a valuable aspect of ML for medical applications, further supporting the possibility of utilizing ML models in the prediction of CRC from the gut microbiome profile of patients.

7.1 Future research

The results presented in this thesis suggest many possible avenues for further research. An important first step would be to evaluate MedPFN more broadly. First and foremost, this includes testing the model on additional compositional, bio-medical datasets. While the experiments were extensive and cross-validation was employed for meaningful results, a more comprehensive evaluation of the proposed methods on more datasets is needed to confirm the result of my research. Furthermore, the training input was limited to 1024 samples in the cross-validation of all experiments. MedPFN's performance on inputs of different context lengths should be measured. It is possible the performance extrapolates to smaller and larger training sizes, as does TabPFN [Hollmann et al., 2023]. MedPFN could also be trained on datasets of varying sizes, or versions could be constructed that are designated for smaller and larger datasets, respectively.

There are several possible approaches for further improving MedPFN's performance. [Rundel et al., 2024] describes employing data valuation techniques for context optimization. For this, samples are ranked through data valuation techniques, such as leave-out-out (LOO). Then, only the samples deemed most important are used in the context of the input. Similarly, patients could be split into groups through some similarity measure and predictions performed only within cohorts. Figure 6.4a shows patients belong to several clusters. Separating those and evaluating the model on the clusters individually might lead to improved classification. This could be especially useful in this setting, as the input context is limited to 1024. The difference in dispersion of the CRC samples between the clinical dataset and my DM prior in Figure 6.4 also suggests that a modified label mapping procedure could create better-suited associations between features and the target variable. A possible approach is to not instantiate one BNN per dataset but work with multiple BNNs to induce a less homogeneous feature-target relationship. Following a similar argument, the K feature-specific variables α_k (see Algorithm 2) can be extended to multiple sets of K features to simulate different microbiome contexts.

The results in Section 6.2.2 as well as Section 6.3.1 indicate that the boost in model performance from the curriculum training regime is not limited to imbalanced datasets. This suggests the class imbalance in synthetic datasets could be an effective mechanism in encouraging PFNs to learn more complex inference patterns outside of the narrow application in compositional, bio-medical data. It is possible that this could be applied or adapted to improve general tabular classifiers, such as TabPFN.

Bibliography

[Aitchison, 2018] Aitchison, J. (2018). The Statistical Analysis of Compositional Data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160.

[Ali et al., 2013] Ali, A., Shamsuddin, S. M., and Ralescu, A. L. (2013). Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*, 5(3):176–204.

[Allali et al., 2017] Allali, I., Arnold, J. W., Roach, J., Cadenas, M. B., Butz, N., Hassan, H. M., Koci, M., Ballou, A., Mendoza, M., Ali, R., et al. (2017). A comparison of sequencing platforms and bioinformatics pipelines for compositional analysis of the gut microbiome. *BMC microbiology*, 17:1–16.

[Ba et al., 2016] Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *CoRR*, abs/1607.06450.

[Borisov et al., 2021] Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. (2021). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

[Chen and Li, 2013] Chen, J. and Li, H. (2013). Variable selection for sparse dirichlet-multinomial regression with an application to microbiome data analysis. *The annals of applied statistics*, 7(1).

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 11, page 785–794. ACM.

[Chen et al., 2012] Chen, W., Liu, F., Ling, Z., Tong, X., and Xiang, C. (2012). Human intestinal lumen and mucosa-associated microbiota in patients with colorectal cancer. *PLOS ONE*, 7(6):1–9.

[Cresci and Bawden,] Cresci, G. A. and Bawden, E. Gut microbiome. *Nutrition in Clinical Practice*, 30(6):734–746.

[Crost et al., 2023] Crost, E. H., Coletto, E., Bell, A., and Juge, N. (2023). Ruminococcus gnavus: friend or foe for human health. *FEMS Microbiology Reviews*, 47(2):fuad014.

[Dave et al., 2012] Dave, M., Higgins, P. D., Middha, S., and Rioux, K. P. (2012). The human gut microbiome: current knowledge, challenges, and future directions. *Translational Research*, 160(4):246–257.

[den Breejen et al., 2024] den Breejen, F., Bae, S., Cha, S., and Yun, S. (2024). Why in-context learning transformers are tabular data classifiers. *CoRR*, abs/2405.13396.

[Dong et al., 2024] Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., and Sui, Z. (2024). A survey on in-context learning. pages 1107–1128.

[Feng et al., 2021] Feng, Y., Zhou, M., and Tong, X. (2021). Imbalanced classification: A paradigm-based review. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 14(5):383–406.

[Galland, 2014] Galland, L. (2014). The gut microbiome and the brain. *Journal of Medicinal Food*, 17(12):1261–1272. PMID: 25402818.

[Gou et al., 2020] Gou, W., Ling, C.-w., He, Y., Jiang, Z., Fu, Y., Xu, F., Miao, Z., Sun, T.-y., Lin, J.-s., Zhu, H.-l., Zhou, H., Chen, Y.-m., and Zheng, J.-S. (2020). Interpretable Machine Learning Framework Reveals Robust Gut Microbiome Features Associated With Type 2 Diabetes. *Diabetes Care*, 44(2):358–366.

[Granados-Romero et al., 2017] Granados-Romero, J. J., Valderrama-Treviño, A. I., Contreras-Flores, E. H., Barrera-Mera, B., Herrera Enríquez, M., Uriarte-Ruíz, K., Ceballos-Villalba, J. C., Estrada-Mata, A. G., Alvarado Rodríguez, C., and Arauz-Peña, G. (2017). Colorectal cancer: a review. *Int J Res Med Sci*, 5(11):4667.

[Grice, 2011] Grice, E., S. J. (2011). The skin microbiome. *Nat Rev Microbiol 9*.

[Grinsztajn et al., 2022] Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.

[Halfvarson et al., 2017] Halfvarson, J., Brislawn, C. J., Lamendella, R., Vázquez-Baeza, Y., Walters, W. A., Bramer, L. M., D'Amato, M., Bonfiglio, F., McDonald, D., Gonzalez, A., McClure, E. E., Dunklebarger, M. F., Knight, R., and Jansson, J. K. (2017). Dynamics of the human gut microbiome in inflammatory bowel disease. *Nature Microbiology*, 2.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

[Hernández Medina et al., 2022] Hernández Medina, R., Kutuzova, S., Nielsen, K. N., Johansen, J., Hansen, L. H., Nielsen, M., and Rasmussen, S. (2022). Machine learning and deep learning applications in microbiome research. *ISME Communications*, 2(1):98.

[Hollmann et al., 2023] Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2023). TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*.

[Holmes et al., 2012] Holmes, I., Harris, K., and Quince, C. (2012). Dirichlet multinomial mixtures: Generative models for microbial metagenomics. *PLOS ONE*, 7(2):1–15.

[Huang and Ling, 2005] Huang, J. and Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310.

[Jafarigol and Trafalis, 2023] Jafarigol, E. and Trafalis, T. (2023). A review of machine learning techniques in imbalanced data and future trends. *arXiv preprint arXiv:2310.07917*.

[Kau et al., 2011] Kau, A., Ahern, P., Griffin, N., Goodman, A., and Gordon, J. (2011). Human nutrition, the gut microbiome and the immune system. *Nature*, pages 327–336.

[Kleinbaum et al., 2002] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., and Klein, M. (2002). *Logistic regression*. Springer.

[Konishi et al., 2022] Konishi, Y., Okumura, S., Matsumoto, T., Itatani, Y., Nishiyama, T., Okazaki, Y., Shibutani, M., Ohtani, N., Nagahara, H., Obama, K., Ohira, M., Sakai, Y., Nagayama, S., and Hara, E. (2022). Development and evaluation of a colorectal cancer screening method using machine learning-based gut microbiota analysis. *Cancer Medicine*, 11(16):3194–3206.

[LeDell and Poirier, 2020] LeDell, E. and Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*.

[Li et al., 2022] Li, P., Luo, H., Ji, B., and Nielsen, J. (2022). Machine learning for data integration in human gut microbiome. *Microbial Cell Factories*, 21(1):241.

[Liu et al., 2017] Liu, J., Williams, B., Frank, D., Dillon, S. M., Wilson, C. C., and Landay, A. L. (2017). Inside Out: HIV, the Gut Microbiome, and the Mucosal Immune System. *The Journal of Immunology*, 198(2):605–614.

[Liu et al., 2022] Liu, W., Fang, X., Zhou, Y., Dou, L., and Dou, T. (2022). Machine learning-based investigation of the relationship between gut microbiome and obesity status. *Microbes and Infection*, 24(2):104892.

[Loshchilov and Hutter, 2017] Loshchilov, I. and Hutter, F. (2017). Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

[Lundberg and Lee, 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

[Maćkiewicz and Ratajczak, 1993] Maćkiewicz, A. and Ratajczak, W. (1993). Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342.

[Manandhar et al., 2021] Manandhar, I., Alimadadi, A., Aryal, S., Munroe, P. B., Joe, B., and Cheng, X. (2021). Gut microbiome-based supervised machine learning for clinical diagnosis of inflammatory bowel diseases. *American Journal of Physiology-Gastrointestinal and Liver Physiology*, 320(3):G328–G337. PMID: 33439104.

[McElfresh et al., 2023] McElfresh, D., Khandagale, S., Valverde, J., Prasad C, V., Ramakrishnan, G., Goldblum, M., and White, C. (2023). When do neural nets outperform boosted trees on tabular data? In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 76336–76369. Curran Associates, Inc.

[McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Grossberger, L. (2018). Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

[Molnar et al., 2020a] Molnar, C., Casalicchio, G., and Bischl, B. (2020a). Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 417–431. Springer.

[Molnar et al., 2020b] Molnar, C., Casalicchio, G., and Bischl, B. (2020b). *Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges*, page 417–431. Springer International Publishing.

[Müller et al., 2023] Müller, A., Curino, C., and Ramakrishnan, R. (2023). Mothernet: A foundational hypernetwork for tabular classification. *CoRR*, abs/2312.08598.

[Müller et al., 2021] Müller, S., Hollmann, N., Pineda-Arango, S., Grabocka, J., and Hutter, F. (2021). Transformers can do bayesian inference. *CoRR*, abs/2112.10510.

[Narii et al., 2024] Narii, N., Zha, L., Sobue, T., Kitamura, T., Komatsu, M., Shimomura, Y., Shiba, S., Mizutani, S., Yamada, T., and Yachida, S. (2024). Intestinal bacteria fluctuating in early-stage colorectal cancer carcinogenesis are associated with diet in healthy adults. *Nutrition and Cancer*, pages 1–8.

[Osman et al., 2021] Osman, M. A., Neoh, H.-m., Ab Mutalib, N.-S., Chin, S.-F., Mazlan, L., Raja Ali, R. A., Zakaria, A. D., Ngiu, C. S., Ang, M. Y., and Jamal, R. (2021). Parvimonas micra, peptostreptococcus stomatis, fusobacterium nucleatum and akkermansia muciniphila as a four-bacteria biomarker panel of colorectal cancer. *Scientific Reports*, 11(1):2925.

[Papoutsoglou et al., 2023] Papoutsoglou, G., Tarazona, S., Lopes, M. B., Klammsteiner, T., Ibrahimi, E., Eckenberger, J., Novielli, P., Tonda, A., Simeon, A., Shigdel, R., Béreux, S., Vitali, G., Tangaro, S., Lahti, L., Temko, A., Claesson, M. J., and Berland, M. (2023). Machine learning approaches in microbiome research: challenges and best practices. *Frontiers in Microbiology*, 14.

[Pawlowsky-Glahn and Egozcue, 2006] Pawlowsky-Glahn, V. and Egozcue, J. J. (2006). Compositional data and their analysis: an introduction. *Geological Society, London, Special Publications*, 264(1):1–10.

[Pearl, 2000] Pearl, J. (2000). *Causality*. Cambridge University Press, New York.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Peirce and Alviña, 2019] Peirce, J. M. and Alviña, K. (2019). The role of inflammation and the gut microbiome in depression and anxiety. *Journal of Neuroscience Research*, 97(10):1223–1241.

[Picard and Ahmed, 2024] Picard, C. and Ahmed, F. (2024). Untrained and unmatched: Fast and accurate zero-training classification for tabular engineering data. *Journal of Mechanical Design*, 146(9):091705.

[Popov et al., 2019] Popov, S., Morozov, S., and Babenko, A. (2019). Neural oblivious decision ensembles for deep learning on tabular data. *CoRR*, abs/1909.06312.

[Prokhorenkova et al., 2018] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6639–6649, Red Hook, NY, USA. Curran Associates Inc.

[Ranjan et al., 2016] Ranjan, R., Rani, A., Metwally, A., McGee, H. S., and Perkins, D. L. (2016). Analysis of the microbiome: Advantages of whole genome shotgun versus 16s amplicon sequencing. *Biochemical and Biophysical Research Communications*, 469(4):967–977.

[Rebersek, 2021] Rebersek, M. (2021). Gut microbiome and its role in colorectal cancer. *BMC cancer*, 21(1):1325.

[Rizzetto et al., 2018] Rizzetto, L., Fava, F., Tuohy, K. M., and Selmi, C. (2018). Connecting the immune system, systemic chronic inflammation and the gut microbiome: The role of sex. *Journal of Autoimmunity*, 92:12–34.

[Ross et al., 2017] Ross, A., Willson, V. L., Ross, A., and Willson, V. L. (2017). One-way anova. *Basic and advanced statistical tests: Writing results sections and creating tables and figures*, pages 21–24.

[Rui Wang, 2021] Rui Wang, Ruqi Tang, B. L. X. M. B. S. H. T. (2021). Gut microbiome, liver immunology, and liver diseases. *Cellular Molecular Immunology*, 18:4–17.

[Rundel et al., 2024] Rundel, D., Kobialka, J., von Crailsheim, C., Feurer, M., Nagler, T., and Rügamer, D. (2024). Interpretable machine learning for tabpfn. In *World Conference on Explainable Artificial Intelligence*, pages 465–476. Springer.

[Shwartz-Ziv and Tishby, 2017] Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810.

[Somepalli et al., 2021] Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., and Goldstein, T. (2021). SAINT: improved neural networks for tabular data via row attention and contrastive pre-training. *CoRR*, abs/2106.01342.

[Su et al., 2022] Su, Q., Liu, Q., Lau, R. I., Zhang, J., Xu, Z., Yeoh, Y. K., Leung, T. W., Tang, W., Zhang, L., Liang, J. Q., et al. (2022). Faecal microbiome-based machine learning for multi-class disease diagnosis. *Nature communications*, 13(1):6818.

[Tolosana-Delgado et al., 2019] Tolosana-Delgado, R., Talebi, H., Khodadadzadeh, M., and Van den Boogaart, K. (2019). On machine learning algorithms and compositional data. In *Proceedings of the 8th International Workshop on Compositional Data Analysis, Terrassa, Spain*, pages 3–8.

[Ulger et al., 2024] Ulger, Y., Delik, A., and Akkız, H. (2024). Gut microbiome and colorectal cancer: discovery of bacterial changes with metagenomics application in turkısh population. *Genes genomics*, 46.

[Van der Maaten and Hinton, 2008] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[Vanschoren et al., 2013] Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60.

[Vapnik, 1991] Vapnik, V. (1991). Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Wade, 2013] Wade, W. G. (2013). The oral microbiome in health and disease. *Pharmacological Research*, 69(1):137–143. SI:Human microbiome and health.

[Wadsworth et al., 2017] Wadsworth, W. D., Argiento, R., Guindani, M., Galloway-Pena, J., Shelburne, S. A., and Vannucci, M. (2017). An integrative bayesian dirichlet-multinomial regression model for the analysis of taxonomic abundances in microbiome data. *BMC bioinformatics*, 18:1–12.

[Wassenaar, 2018] Wassenaar, T. M. (2018). E. coli and colorectal cancer: a complex relationship that deserves a critical mindset. *Critical Reviews in Microbiology*, 44(5):619–632. PMID: 29909724.

[Williams and Rasmussen, 2006] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

[Wu et al., 2023] Wu, X., Tang, Z., Zhao, R., Yusi, W., Wang, X., Liu, S., and Zou, H. (2023). Taxonomic and functional profiling of fecal metagenomes for the early detection of colorectal cancer. *Frontiers in oncology*, 13:1218056.

[Yerke et al., 2024] Yerke, A., Fry Brumit, D., and Fodor, A. A. (2024). Proportion-based normalizations outperform compositional data transformations in machine learning applications. volume 12, page 45. Springer.

[Zou et al., 2016] Zou, Q., Xie, S., Lin, Z., Wu, M., and Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. *Big Data Research*, 5:2–8. Big data analytics and applications.