Deep Learning in classification of common Carp intestine classification

Marek Sztuka

May 2025

Abstract

Deep Learning in classification of common Carp intestine classification

Deep learning methods can overcome many challenges of high-dimensional, sparse microbiome data. This study, aimed to apply and compare feedforward and convolutional neural networks combined with two dimensionality-reduction techniques; PCA and autoencoder embeddings, in order to classify probiotic treatments from common carp (Cyprinus carpio) gut microbiomes. Transformed via CLR abundance tables from 125 samples served as a dataset for evaluation of six configurations (raw-FNN, raw-CNN, PCA-FNN, PCA-CNN, AE-FNN, AE-CNN) under five-, three-, and two-class scenarios. Models were tuned via 5-fold cross-validation and trained with early stopping. Baselines included random assignment and XGBoost . Results show that AE-FNN achieved the highest accuracy (84% in five classes), outperforming all other methods. PCA-FNN and raw-FNN also exceeded baselines in multi-class tasks, while CNNs; especially without reduction, underperformed, reflecting unstructured feature spaces. FNNs matched or surpassed baselines across scenarios, demonstrating robustness on compositional data. Autoencoder embeddings captured nonlinear patterns that improved the most complex classification task. Future work should integrate feature-importance analyses to reveal key taxa, incorporate environmental and host metadata, and explore larger datasets and advanced architectures (e.g., graph or transformer models) to enhance predictive accuracy and biological insight.

Key Words: Neural Networks, Gut microbiota, autoencoder, Dimensionality reduction, Classification, Machine learning in aquaculture

Chapter 1

INTRODUCTION

1.1 Introduction

The biological role of the microbiome Microbial communities, collectively known as the microbiome, inhabit every environment on Earth, from soil to the human gut. As knowledge in this field grows, an increasing number of studies focus on microbiome analysis. Most of these investigations aim to associate microbiome composition with host phenotypes (Blaser, 2014; Gilbert et al., 2018), or attempt to define a "healthy" microbiome and examine how deviations from this state can affect host well-being (Lloyd-Price et al., 2016).

In aquatic organisms, the microbiome is critically involved in nutrient absorption, immune defense, and overall physiological balance (Sehnal et al., 2021). Disruptions to this delicate community can impact fish growth, disease resistance, and the overall sustainability of aquaculture systems (Kanika et al., 2025).

Probiotics are live microorganisms administered to the host or its environment. They have emerged as promising tools to modulate microbiota composition (Kristensen et al., 2016). In aquaculture, probiotic supplementation typically involves administering these microbes through feed or directly into the water. In fish, such supplementation can (i) outcompete pathogens for ecological niches, (ii) secrete beneficial metabolites, and (iii) stimulate the host immune system (Nayak, 2010).

The Common Carp *(Cyprinus carpio)*, one of the world's most widely farmed fish species, is a prime subject for microbiome-based interventions. Understanding its intestinal microbiome and the effect of probiotic supplementation can inform strategies to improve feed efficiency and disease resistance, contributing to more sustainable aquaculture practices. Recent studies have demonstrated that specific bacterial taxa, such as members of the *Lactobacillaceae* and *Vibrionaceae* families, are associated with improved digestion and increased resistance to pathogens in carp (Lee et al., 2019; Zheng et al., 2020).

1.2 Bioinformatic challanges in microbiome analysis

Despite growing interest, analyzing microbiome data remains challenging. Such datasets are inherently high-dimensional (with hundreds of bacterial taxa), sparse (with many zero-abundance counts), and compositional (where only relative abundances are meaningful). Traditional statistical tests for differential abundance typically analyze each taxon independently, often leading to inflated false discovery rates and neglecting important co-occurrence structures (Gloor et al., 2017; Lin & Peddada, 2020).

To address these limitations, researchers increasingly rely on dimensionality reduction techniques (DRTs), such as Principal Component Analysis (PCA) (Pearson, 1901) and nonlinear embeddings like autoencoders (Liou et al., 2014). These approaches project high-dimensional microbiome data into lower-dimensional latent spaces, making them easier to model and interpret. However, a common drawback of such methods is reduced interpretability; it becomes difficult to trace which bacterial taxa most strongly influence classification outcomes.

1.3 Machine learning in microbiome analysis

Deep Learning (DL) models such as neural networks, which were first introduced by McCulloch and Pitts (1943) in 1943, have recently gained widespread popularity, largely due to advances in computational power and algorithmic design. They now play critical roles in everyday applications, from image recognition using architectures like ResNet-50 (Koonce, 2021) to natural language processing systems like Google Gemini (Team et al., 2025) and ChatGPT by OpenAI (OpenAI, 2023).

In scientific research, DL is increasingly used in practically all aspects of data analysis. Tools such as BEEtag (Crall et al., 2015) and DANNCE (Karashchuk et al., 2021) enable precise tracking of animal behavior. In genomics, models like CNNScoreVariants from the GATK toolkit apply convolutional neural networks to variant scoring (der Auwera & O'Connor, 2020), Alpha Fold (Jumper et al., 2021) is well known algorythm that predicts protein structure utilizing modified transformer architecture, while Dragen software (Behera et al., 2024) allows for comperhensive genome analysis and variant detection while using machine learning to improve single nucleotide variant calling.

One of the key advantages of deep learning architectures over traditional machine learning models (e.g., decision trees or XGBoost) lies in their ability to integrate embedding layers and apply regularization techniques such as dropout. These features are especially beneficial for handling sparse and noisy microbiome data. Additionally, DL models can learn complex, nonlinear relationships between co-occurring bacterial taxa, potentially revealing biologically meaningful patterns that are not accessible through conventional statistical methods. Given these advances, deep learning offers a promising framework for microbiome-based classification tasks.

1.4 Motivation

As both microbiome research and deep learning continue to expand, the main goal of this study was to go beyond traditional approaches that test the differential abundance of individual bacterial families. Instead, it aimed to model the abundance of all bacterial families simultaneously, treating the overall microbial community as a set characteristic for each individual. From a bioinformatic perspective, the study benchmarked random and XGBoost baselines against deep learning (DL) models across multiple class-split scenarios (5, 3, and 2 classes). It also assessed how dimensionality reduction techniques, specifically PCA and autoencoder embeddings, influenced model performance and convergence. The practical objective was to develop a DL-based classifier capable of assigning fish to their respective ponds based on differences in their gut microbiome composition.

Chapter 2

MATERIALS

2.1 Data

The data set originated from the experiment aimed to investigate whether probiotic supplementation of water and feed alters the composition of microbial communities of water, sediment and fish gut. This experiment was carried out by Theta biostatistic group. As there is no publication with description of the methodology of acquiring the data, the following section is dedicated to description of this process.

2.1.1 Data acquisition

The Common carp *(Cyprinus carpio)* was used as the experimental organism. The experiment consisted of five experimental setups which differed in supplementation scheme (table experimental setup), that involved the **control group** - without any supplementation, **group 1** - without feed supplementation and water supplement **S1**, **group 2** - with feed supplementation **S2** and with water supplementation **S1**, **group 3** without feed supplementation and with **S3** water supplementation and **group 4** - with **S3** feed and water supplementation. Each group was represented by five ponds with unified environmental conditions, the experiment lasted for five months, from May and to October. At the end of the experiment five samples of fish gut were collected from each pond resulting in 125 samples.

2.1.2 Microbiome sequencing and taxonomic annotation

For the S16 ribosomal subunit (16S rRNA gene) sequencing (Johnson et al., 2019) DNA isolation was performed using commercial DNA isolation kit QIAamp PowerFecal Pro. Two hypervariable regions (v3 and v4) of the gene were sequenced following recommended by Illumina procedure and using Illumina Nova Seq 6000 (Modi et al., 2021) in pair-end mode of 2x250 bp (two fragments of length 250).

Data was then pre-processed and taxonomically annotated using Quantitative Insights Into Microbial Ecology 2 (QIIME2) software (Bolyen et al., 2019), that step yielded in-

dividual so-called feature tables for each pond which contained absolute abundance of identified bacteria families in each sample, those tables were then concatenated using the Numpy library (Harris et al., 2020) and Bash (GNU, 2007) into one table containing abundances from gut bacterial families of all fish. Additionally it is worth mentioning that one of the samples from the control group was missing but as this didn't cause class imbalance problems it wasn't dealt with in any way. **The final table consisted of 126 columns corresponding to bacteria families and 124 rows corresponding to fish. This data set formed the basis of all analyses carried out within the frame of this thesis.**

2.2 Computational setup

2.2.1 Hardware

All computations were carried out on a laptop equipped with an Intel(R) Core(TM) Ultra 5 125H CPU, featuring 14 cores and 18 threads. This includes 4 performance cores, each capable of running 2 threads at a maximum frequency of 4.5 GHz, which handles most of the computational tasks. The remaining 8 efficiency cores were responsible for data management and background operations, supporting smooth multitasking.

This hybrid architecture allowed for real-time computation, with each model run completing in under 5 minutes on the local machine. The system was also equipped with 16 GB of RAM, which was sufficient for managing the dataset and training the deep learning models without requiring external computational resources.

2.2.2 Software

All analyses in this study were conducted using Python v3.12 (Rossum & Drake, 2009) as the primary programming language. Several Python packages were used throughout the project: (i) Numpy (Harris et al., 2020) and (ii) Pandas (McKinney et al., 2010) were used for data handling, including preprocessing, input transformation, and output parsing. These libraries facilitated efficient manipulation of the bacterial abundance tables.(iii) Scikit learn (Pedregosa et al., 2011) was employed for splitting the dataset into validation and test sets, performing cross-validation, normalization, and dimensionality reduction via PCA. It also provided evaluation metrics such as accuracy and confusion matrices.(iv) Tensorflow (Abadi et al., 2016) served as the backend framework for creating, training, and testing the deep learning models. TensorFlow managed data in the form of tensors during model operations. On top of this (v) Keras API (Chollet et al., 2015) was used as the front-end interface for building neural networks and applying utilities like one-hot encoding. For the purpose of implementing the baseline algorithm (vi) XGBoost (T. Chen & Guestrin, 2016) package was used. For visualizations—such as plotting accuracy metrics, confusion matrices, and model comparisons - (vii) Matplotlib (Hunter, 2007) was used. All development and analysis were carried out in Jupyter Notebooks (Kluyver et al., 2016) allowing for a reproducible and well-documented computational workflow.

Chapter 3

METHODS

3.1 Data pre-processing and input transformation

The first step in analysis was inspection of the dataset mainly focused on the number of missing values. Missing values either represent bacterial families absent in a given community or families that were abundant but below the detectable level. Those values were replaced with zeros. Statistically, such data frame structure is called zero inflated count data (Gloor et al., 2017) and in traditional modelling requires dedicated statistical methodology (Feng, 2021). Conversely, DL architectures provide a robust approach to handling data with missing values (Haghani et al., 2017). Another important aspect of data inspection was the variability across bacterial family abundance, that was originally expressed as sequence read counts. Large disparities in abundance can cause machine learning models to disproportionately focus on highly abundant bacterial families, potentially ignoring less prevalent but biologically relevant ones, moreover the raw read count is not comparable across samples. To address those issues, data normalisation of the raw data was applied. In this study, the Centered Log-Ratio (CLR) (eq. 3.1) transformation was used.

$$CLR(x_i) = ln\frac{x_i}{g(X)} \tag{3.1}$$

The CLR transformation for a given observation x_i is calculated as the natural logarithm of the ratio between that observation and the geometric mean of all observations from the same animal. In this equation, x_i represents the abundance of a single bacterial feature, while $g(X)$ denotes the geometric mean of all values in the corresponding sample vector X. This transformation is applied individually to each sample across the entire dataset to normalize the data and reduce compositional bias. This is a commonly used method for handling count microbiome data, it was introduced by Aitchison Aitchison (1982) as he realized in his study that the underlying relations in such data can be captured by treating abundances as ratios (Gloor et al., 2016). This ensures that the relative abundance values are preserved while mitigating the effect of large range differences in raw read counts.

Further data processing, prior to the application of machine learning methods, included splitting the full dataset into a training and a validation (test) set. The split was performed

randomly, with the training set comprising approximately 80% of the full dataset, including microbiome data from 99 individual fish. The validation set consisted of the remaining 20%, corresponding to 25 samples. To ensure fair model evaluation, the split was stratified, meaning that the original proportions of class labels were kept in both subsets. Additionally, the distribution of samples originating from different ponds was random, preventing any systematic bias related to environmental conditions.

3.2 Crossvalidation

To enhance the reliability of the training process, the training set was further divided using 5-fold cross-validation. In this method, the dataset is split into five equal parts (folds), and the model is trained five times — each time using four folds for training and one for validation. This ensures that each sample is used once for validation and four times for training, helping to reduce bias and increase the robustness of the results.

3.3 Class reduction

Given that the initial **5-class split** may not accurately reflect true biological relationships, additional class groupings were introduced, based on the assumption that feed supplementation has a greater impact on the intestinal microbiome. Specifically, we additionally tested: (i) A **3-class split**, where the **control group** remained unchanged, while **group 1** was merged with **group 3**, and **group 2** with **group 4**, based on similarities in supplementation. (ii) A **2-class split**, where the **control**, **group 1**, and **group 3** were grouped into one class, and **group 2** with **group 4** formed the other class.

3.4 Dimensionality reduction

After data preprocessing, the next step was to explore various dimensionality reduction techniques (DRT). The application of dimensionality reduction of the original data was driven by the following ideas:

- Due to the low number of samples (i.e. ponds) and zero-inflated structure of the data, it may be beneficial to reduce the number of features that serve as input for the deep learning classification architectures, without compromising on information loss, since the reduced number of features still represents variation of the original data well, to perform classification with high accuracy.
- The reduced data representation prevents fitting artificial patterns to residual (i.e. nonexperimental) variation present in the analyzed samples, which is especially important providing a relatively low accuracy and results associated with environmental samples as compared to e.g. tissue or cultured samples. The nonlinear rep-

resentation of the original data may allow for better pattern representation than the raw measurements themselves. (Durán et al., 2021)

In the study, two approaches of reduced data representation were used:

- PCA (Pearson, 1901) is a widely used dimensionality reduction technique that transforms the original dataset into a set of orthogonal components called principal components (PCs). It has also been used in microbiome data applications. Each PC captures a portion of the total variance in the data, and the first few components typically retain the most meaningful structure. By selecting a subset of these components, one can significantly reduce the number of features while still preserving most of the original variance. In this study, PCA was implemented using the scikit-learn package. The transformation model was fitted to the training dataset and subsequently applied to both the training and test datasets to ensure a consistent representation. PCA was run iteratively with an increasing number of components, with the initial target threshold for the cumulative explained variance set at 90%.

- Deep Learning based embedding based on Autoencoders (AE) (Berahmand et al., 2024) The AE architecture is a type of unsupervised artificial neural network used for learning efficient, compressed representations of input data. It consists of two main components: an encoder, which maps the input data into a lower-dimensional latent space - embeddings vector, and a decoder, which attempts to reconstruct the original input from this compressed representation (fig. 3.1). The loss function was defined as the highest similarity between the original and the reconstructed input. In this study, the AE was trained by minimizing the Mean Squared Error (MSE) (eq. 3.2). Where x_i denotes single observation from fish i, while $\hat{x}_i$ is same observation reconstructed by AE. The AE architecture used in this study consisted of a symmetrical encoder and decoder, each composed of three dense (fully connected) layers with linear activation functions. The encoder included layers with 64, 32, and 19 nodes, resulting in an embedding vector of length 19 for each input sample. This produced an output matrix of shape $n\times19$, where $n = 124$ (the number of samples). The model was trained on the validation set for 2000 epochs. A high number of training epochs is often acceptable and even beneficial, when training autoencoders, as overfitting is not necessarily undesirable in this context. Since the goal is to accurately reconstruct the input data, models trained extensively can better capture the structure of the data. As such, no early stopping was applied; convergence was assessed manually. The ADAM optimizer was used due to its robust and efficient performance across many deep learning tasks. Various autoencoder architectures were tested, differing in the number of layers and nodes per layer. The configuration described above yielded the best reconstruction quality and was selected for downstream analysis. The matrix of embeddings corresponding to the final AE model was used as the reduced representation of the input data set.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{3.2}$$

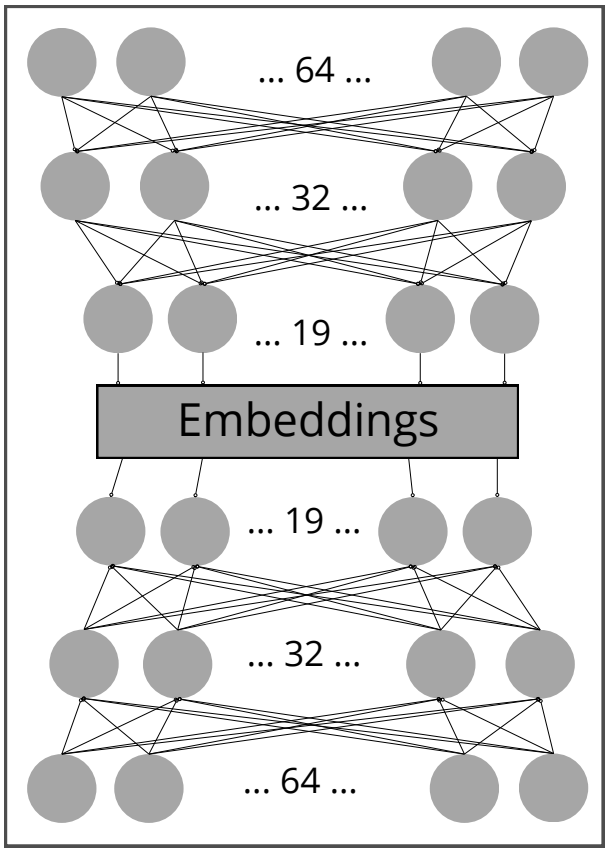


Figure 3.1: Autoencoder architecture utilized in this study for dimensionality reduction purposes, it composes of Encoder (top) with 3 dense (fully connected) layers (64,32 and 19 nodes) and Decoder (bottom) with symetrical dense layer layout. No activation function was used.

3.5 Neural network architectures

The next step in our analysis focused on evaluating various neural network (NN) architectures for microbiome classification. Neural networks are computational models commonly used for classification and prediction tasks. They operate by propagating input data through a network of interconnected units, called neurons. Each neuron applies an activation function to its inputs, and the model learns by adjusting the weights of these connections through a process known as backpropagation during training (Rumelhart et al., 1986). In this study two types of architectures were utilized, a Feed Forward Neural Network (FNN) (Bebis & Georgiopoulos, 1994) and Convolutional Neural Network (CNN) (Fukushima, 1979). Both types of networks had a common set of parameters: Rectified Linear Unit (ReLU) was the activation function for dense layers and smooth approximation of one-hot arg max (Softmax) was used for classification purposes at the output layer. Adaptive Moment Estimation (ADAM) (Kingma & Ba, 2017) was used to minimize the Categorical Crossentropy loss function. There were also common overfitting prevention

measures applied to both architectures: weight regularizer L2 with high regularization factor $\lambda = 0.01$ eq. 3.3 and Dropout Layers which randomly remove selected edges in the network.

$$R_{L2}(W) = \lambda \sum_{i=1}^{N} w_i^2 \tag{3.3}$$

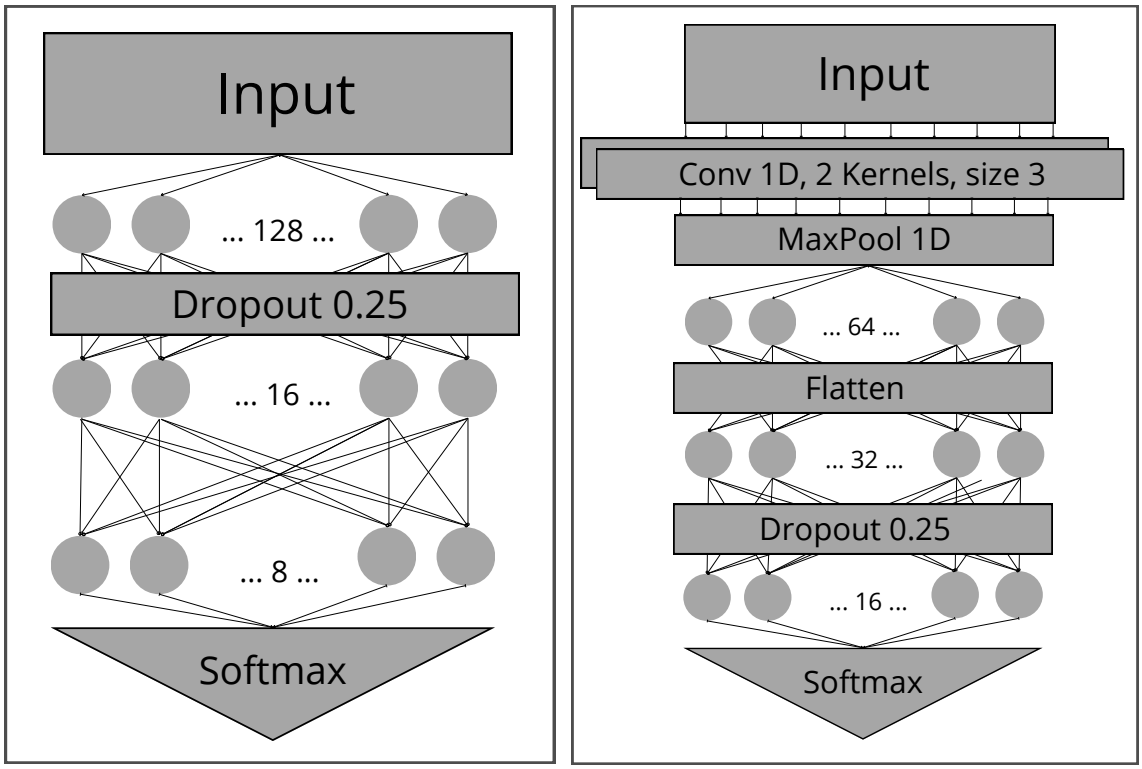
In eq. 3.3; w_i represents individual weight from the layer, N is the total number of weights in that layer and λ is regularization factor.

3.5.1 FNN

The general FNN model consisted of 3 dense layers with consecutively decreasing number of nodes (128, 64, 8) followed by output layer with number of nodes corresponding to the given class split (5,3 or 2). The dropout layer was placed after the first dense layer with a dropout rate of 0.25. fig. 3.2a.

3.5.2 CNN

CNNs are traditionally used in image analysis. The hypothesis underlying this study was that they might extract local abundance patterns from microbiome data. Particularly when paired with dimensionality reduction techniques. The key difference in this approach compared to FNN, lies in the use of convolutional layers, which are capable of detecting and summarizing spatial or structural patterns across the input features. The applied CNN architecture utilized a 1D convolutional layer (Kiranyaz et al., 2021) to the input. This layer employed two convolutional kernels, each with a size of 3, allowing them to slide across the input data and attempt to identify local structural patterns between features. To enable this, the input data was formatted such that each individual animal's bacterial abundance represented a one-dimensional "image". This transformation allowed the convolutional filters to process the data sequentially. Although the original dataset did not have any inherent spatial or sequential structure, since the features (bacterial families) were not ordered, it was hypothesized that DRTs such as PCA or autoencoder embeddings create patterns that are easier to estimate by a convolutional layer. Following the convolutional layer, a MaxPooling1D layer with a pool size of 4 was applied. This operation downsampled the data by retaining only the maximum value within each window, reducing its dimensionality and highlighting the most dominant local patterns. The next layers consisted of a Dense layer with 64 units, followed by a Flatten layer to reshape the output for further processing by dense layers. The final part of the architecture included two additional Dense layers with 32 and 16 nodes, respectively, with a Dropout layer (with a dropout rate of 0.25) in between to mitigate overfitting. The network concluded with a Softmax layer for multi-class classification fig. 3.2b.



(a) Dense architecture composes of 3 dense layers with (128,64 and 8) nodes, followed by an output layer with number of nodes corresponding to given class split (5,3 or 2). The dropout layer was placed after the first dense layer with 0.25 dropout rate.

(b) Convolutional architecture consisting of 1D convolutional layer which employed two convolutional kernels, each with a size of 3. This was followed by: one MaxPool1D layer, dense layer with 64 nodes, Flatten layer and then two another dense layers with (32, 16) nodes, with Dropout (0.25) inbetween. The output layer was Softmax for classification with number of nodes coresponding to class split (5,3 or 2)

Figure 3.2: DL architectures: (a) Feedforward Neural Network (FNN), (b) Convolutional Neural Network (CNN).

3.6 Hyperparameter tuning and architectural variants

To optimize model performance, extensive hyperparameter tuning was conducted for both FNN and CNN architectures. The goal was to identify the most effective configurations for each dimensionality reduction setting. Although not all tested configurations are reported in the main results, a wide range of architectural variants were explored:

3.6.1 Feedforward approaches:

- **Dense Layers:** 1 to 3 layers with activation functions including sigmoid, ReLU, and linear. The number of nodes tested per layer ranged from 2048 to 16.
- **Dropout:** Dropout layers were inserted with rates ranging from 0.0 to 0.5 to control overfitting.
- **Dense Layers:** 1 to 3 layers after dropout with activation functions including sigmoid, ReLU, and linear. The number of nodes tested per layer ranged from 1024 to 8.
- **Output Layer:** A final softmax layer was used for classification.

3.6.2 Convolutional approaches:

- **Convolutional Layers:** 1 to 2 one-dimensional convolutional layers were tested with kernel sizes ranging from 2 to 126 and filter counts from 1 to 10. Activation functions included sigmoid, ReLU, and linear.
- **Pooling:** 1D MaxPooling layers were included following convolution.
- **Flattening and Dense Layers:** After and before flattening, 1 to 3 dense layers were tested with node counts from 2048 to 8, again using the same set of activation functions. Dropout was applied at different stages with rates from 0.0 to 0.5.
- **Output Layer:** A softmax layer was used for classification output.

3.6.3 Input transformations

Different input normalization techniques were also tested, including standard normalization and min-max scaling, to examine their effect on training stability and accuracy.

In the next step neural network architectures were individually tuned and validated using both the original dataset and its dimensionally reduced versions (via PCA and AE). This resulted in a total of six DRT-NN configurations: X-FNN, X-CNN, PCA-FNN, PCA-CNN, AE-FNN, AE-CNN. Where X denotes that no DRT was applied.

3.7 Model performance

Model performance was primarily evaluated using accuracy as the main classification metric. Accuracy is defined as the ratio of correctly predicted class labels T to the total number of predictions made, which corresponds to the size of the test set n as shown in eq. 3.4

$$accuracy = \frac{T}{n} \tag{3.4}$$

For a more precise evaluation, Confusion Matrices were examined for a more detailed analysis of classification error patterns. These matrices provide insight into how well each class was predicted, highlighting specific misclassifications. This level of detail is particularly useful in the biological context of this study, as it allows for the identification of systematic patterns in model errors. Such patterns may reveal overlap between groups or suggest biologically meaningful class reductions based on the similarity of microbiome profiles under certain supplementation schemes. Models were also inspected with regard to healthy model training expressed by decreasing the loss function over training epochs, overfitting, and need for computational resources.

To ensure robustness and generalizability of the results, especially important in smaller datasets where variability between training runs is common. Each model configuration was tuned using the previously described 5-fold cross-validation strategy. This procedure enabled consistent performance evaluation across training subsets and helped mitigate overfitting. During this tuning phase, each model was trained for 300 epochs per fold.

3.8 Classification baselines

Given the absence of definitive knowledge regarding the true impact of probiotic supplementation on bacterial family abundance between ponds. For the purpose of this study, two baseline classifications were used. They allowed for the assessment of the relative quality of the Deep Learning based classifiers proposed for the analysed data set:

- The classification accuracy assuming the null hypothesis of no effect of probiotic supplementation, expressed by a random assignment of ponds to classes. For example, for 5 class scenarios, the random pond assignment would result in accuracy of 0.2.

- The classification accuracy based on the variation in real data was assessed by the XGBoost, tree based classifier that represents a robust classification approach (Wu et al., 2021). The XGBoost classification was implemented in the XGBoost package and conducted using the maximum tree depth of 3 and eta parameter which controls contribution of new trees and thus learning rate of the model was set to 0.1. The classifier was trained for 100 iterations on the validation set and then tested on the test set.

3.9 Final model testing and evaluation

The final step of the analysis was to test previously tuned models on unseen data. Each model configuration was trained on the entire validation set for up to 500 epochs. As a part of overfitting prevention measures, early stopping callback was introduced. This technique monitors the loss on the test data and halts training when it begins to diverge significantly from the training loss, indicating that the model has stopped generalizing well. After training, each model was evaluated on its corresponding test set (matched to its dimensionality reduction technique, if used).

Chapter 4

RESULTS

4.1 Data inspection

4.1.1 Missing values

The initial step of data pre-processing involved inspecting the dataset for missing values. There was a high proportion of zero-abundance entries across the data set. In particular, over 80 bacterial families (63%) were not abundant in at least 99 out of 124 individuals. Notably, 39 bacterial families (31%) were detected in only five individuals, which indicates that these bacteria were observed in only one pond. The distribution of missing values across bacterial families was visualized in fig. 4.1.

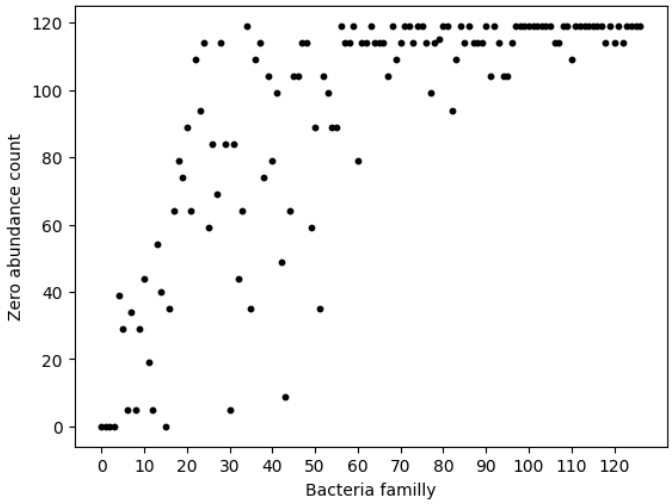


Figure 4.1: Plot showing the distribution of zero-abundance counts across bacterial families. The X-axis represents individual features (bacterial families), while the Y-axis indicates the number of samples in which each family was absent. The high concentration of zeros suggests a sparse dataset, with many taxa detected only in a few individuals or single ponds.

4.1.2 Distribution of raw abundances

The distribution of raw read counts was highly imbalanced across individual fish and bacterial families as can be seen on fig. 4.2 where range of values was demonstrated on logarithmic scale. While the majority of abundances (88 families, 70%) — did not exceed 100 read counts, some features demonstrated extremely high maximum read count values. In particular, 3 families (2.4%) exhibited ranges exceeding 10,000.

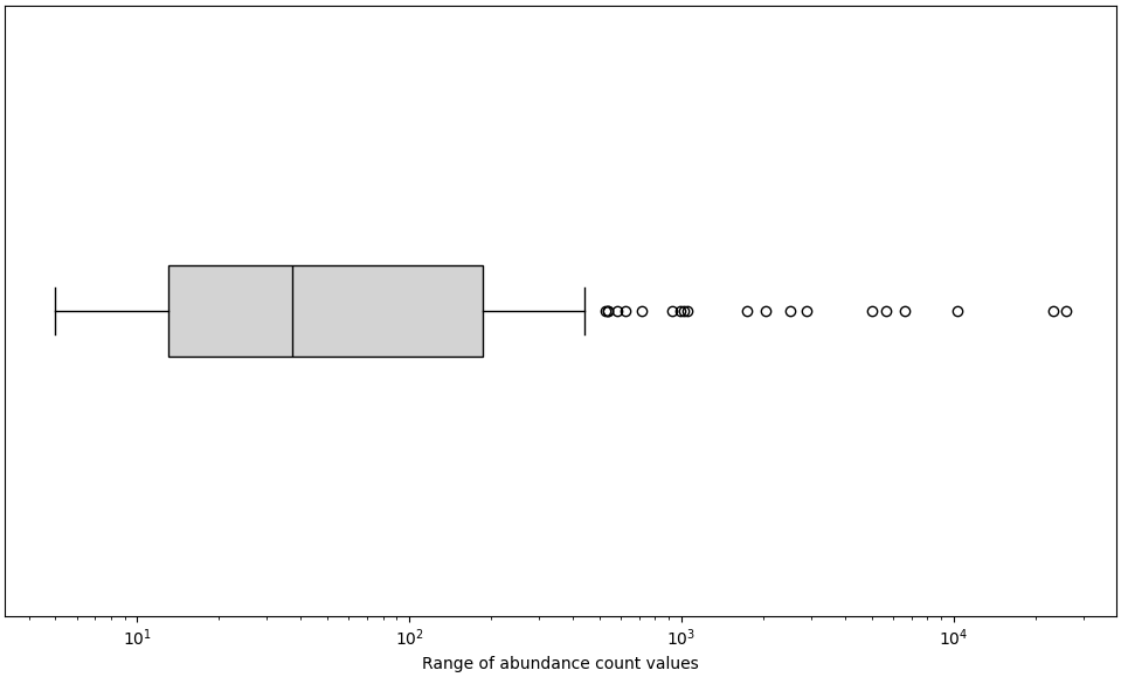


Figure 4.2: Boxplot illustrating the range of bacterial abundance values across features. The X-axis represents the range of read counts on a logarithmic scale. Most families had relatively low abundance variability, but a few displayed extreme ranges exceeding 10,000 reads.

4.2 Dimensionality reduction

4.2.1 PCA

After fitting the PCA transformation model to the training data, it was determined that 22 principal components (PCs) were required to explain 90% of the total variance (fig. 4.3). To further identify the optimal number of components for classification, a range of PCs varying from 15 to 30 was used within the same deep learning architecture. Among these, the data representation with 17 principal components yielded the best classification performance (accuracy of 0.59) for 5 classes on the validation set, and was selected for subsequent model comparisons.

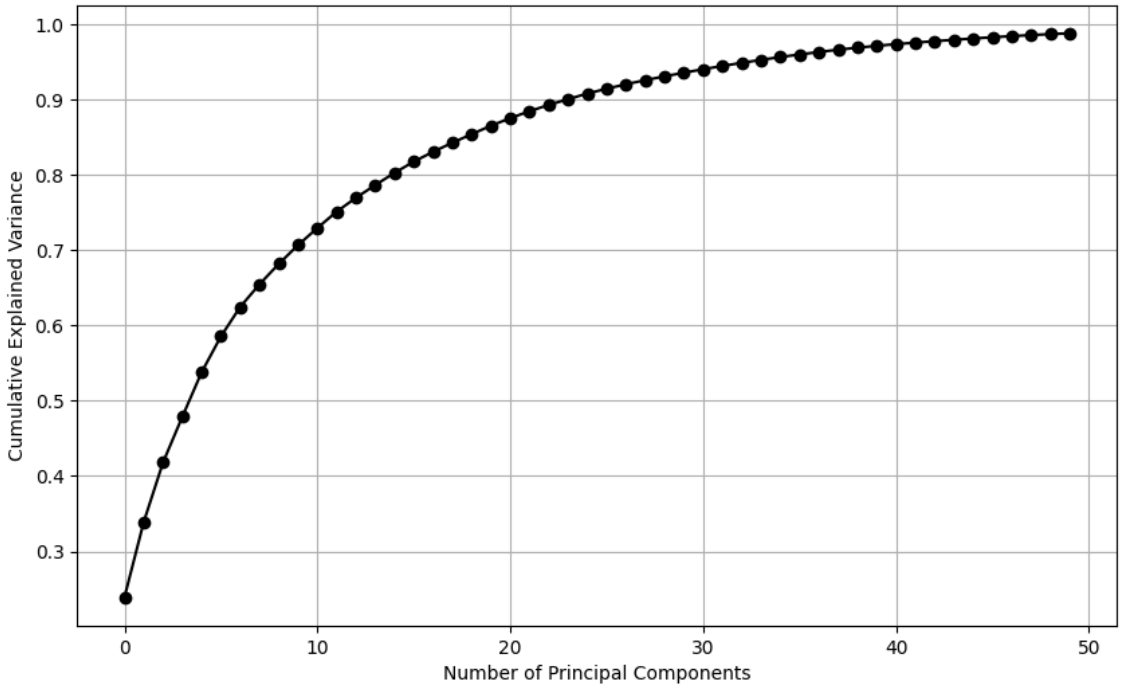
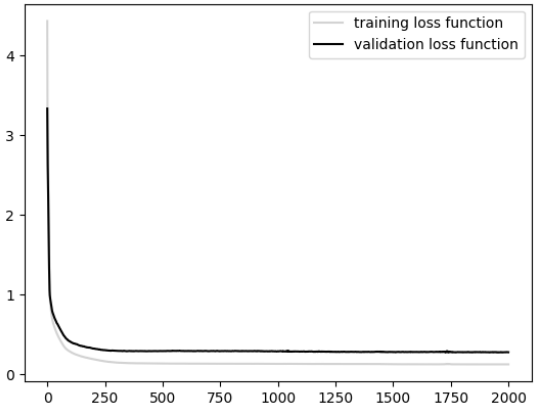


Figure 4.3: Scree plot showing cumulative explained variance (Y-axis) as a function of the number of principal components (X-axis). A 90% variance threshold was reached with 22 components, indicating sufficient dimensionality for preserving most of the dataset's variability.

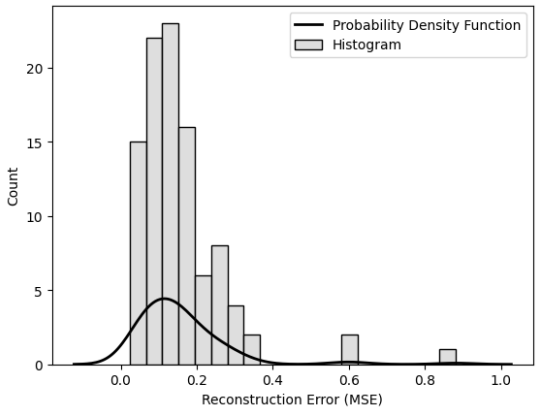
4.2.2 AE

An AE model was trained for 2000 epochs. As shown in fig. 4.4a, the loss function plateaued after 500 epochs, indicating that further training had minimal impact on reconstruction quality. To evaluate the reconstruction performance, the mean squared error

(MSE) was calculated between the original and reconstructed abundance of c=bacterial families for each individual. The distribution of MSE is shown in fig. 4.4b. The mean MSE across all samples was 0.29. The embedding vector size of 19 was selected based on initial testing, where it achieved the highest validation accuracy (0.52) in the 5-class classification split.



(a) Loss function during AE training. The Y-axis represents loss values, and the X-axis indicates training epochs. Both training and validation loss curves plateaued after 500 epochs.



(b) Histogram with the distribution of reconstruction MSE values across samples. The X-axis represents MSE, and the Y-axis shows the number of samples with each MSE range. Most errors fall between 0 and 0.4, with a few exceeding 0.6. A probability density function was overlaid.

Figure 4.4: Autoencoder training evaluation: (a) loss function over epochs, (b) reconstruction error distribution.

4.3 DL Classification

Following dimensionality reduction, the next step involved training and evaluating the developed classifiers.

4.3.1 5-class split

In a 5-class split, the goal was to differentiate between all experimental groups: control, group 1, group 2, group 3 and group 4. The number of training epochs varied between 100 and 500, depending on the configuration.

The baseline, random class assignment yielded the expected accuracy of 0.20. The XGBoost algorithm, serving as the benchmark, resulted in the accuracy of 0.56 on the test set. Across the DRT configurations, validation accuracy scores varied substantially

(fig. 4.5 and tab. **??**). The lowest mean validation accuracy of 0.32 (±0.05) was observed for the PCA–CNN configuration, whereas the highest mean validation accuracy of 0.60 (±0.10) was achieved by the PCA–FNN model, followed by the raw–FNN model (i.e. without input dimensionality reduction) which resulted in a mean accuracy of 0.58 (±0.06). Standard deviations of validation accuracy ranged from 0.05 for PCA–CNN to 0.12 for raw–CNN configurations, indicating varying levels of classification robustness across models.

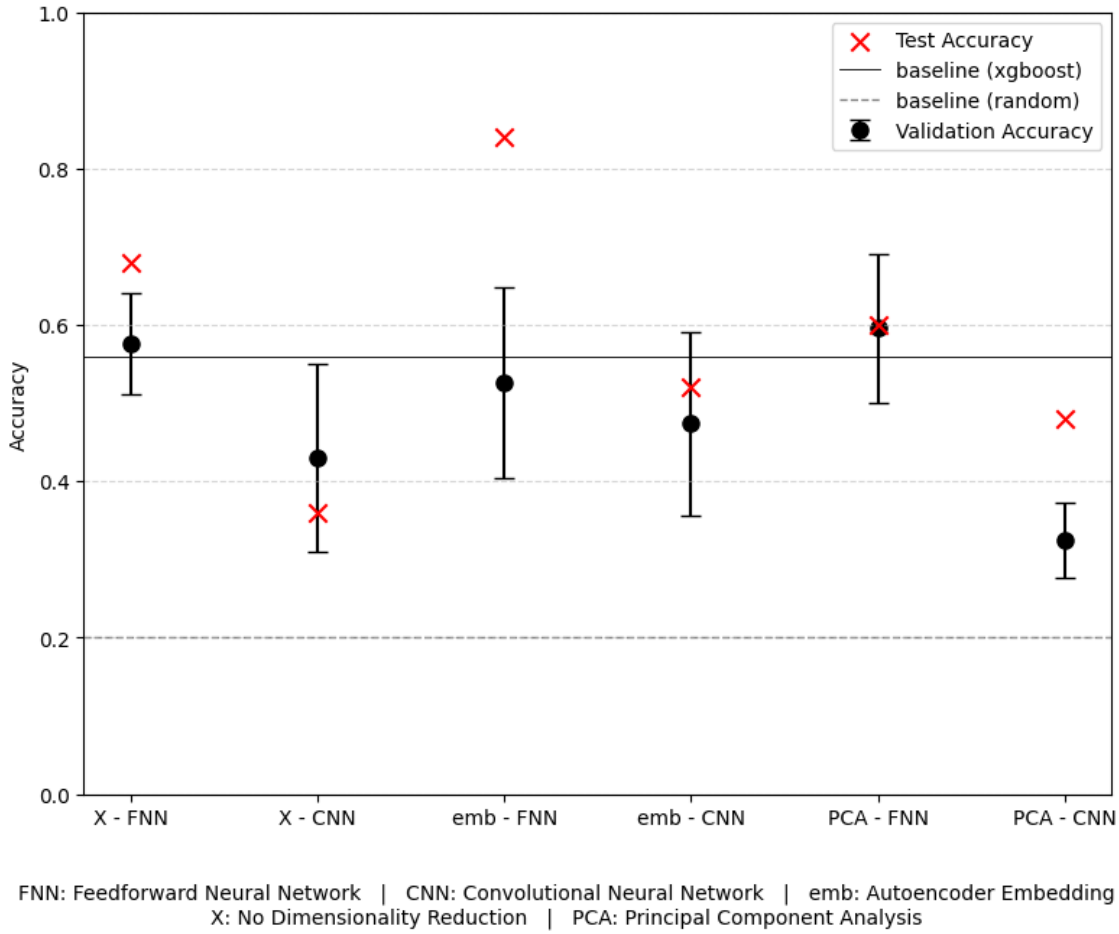


Figure 4.5: 5-class split plot showing classification accuracy for all six DRT–NN configurations (represented on X-axis). The Y-axis shows accuracy scores. The red 'X' marks indicate test set accuracy, and the black whiskers represent mean validation accuracy with standard deviation from 5-fold cross-validation. The black horizontal line denotes the XGBoost baseline accuracy (0.56), while the gray line marks the random classification baseline (0.20). Among tested configurations, PCA–FNN achieved the highest validation accuracy (0.60 ± 0.10), followed by raw–FNN (0.58 ± 0.06). The lowest validation performance was observed for PCA–CNN (0.32 ± 0.05).

4.3.2 3-class split

A 3-class classification scenario was also evaluated to compare: (i) **Class 0** represented the Control group, (ii) **Class 1** combined Group 1 and Group 3, (iii) **Class 2** combined Group 2 and Group 4. For 3 classes, the random baseline is defined by the accuracy of 0.33. The XGBoost algorithm yielded an accuracy of 0.64. The mean accuracy of validation sub-samples was more consistent across applied models than in the 5-class scenario, ranging from 0.57 for PCA–CNN to 0.70 for the raw–CNN configuration and 0.69 for PCA–FNN. The standard deviation of validation accuracies was also lower in the 3-class setting, ranging between 0.08 and 0.09 for most approaches, except for the raw–CNN with the lowest deviation of 0.03. Test accuracy results were also more uniform across configurations, ranging from 0.72 for both PCA–CNN and plain CNN, to a maximum of 0.84 for the raw–FNN model, followed by the emb–FNN at 0.80 (fig. 4.6 and tab. **??**).

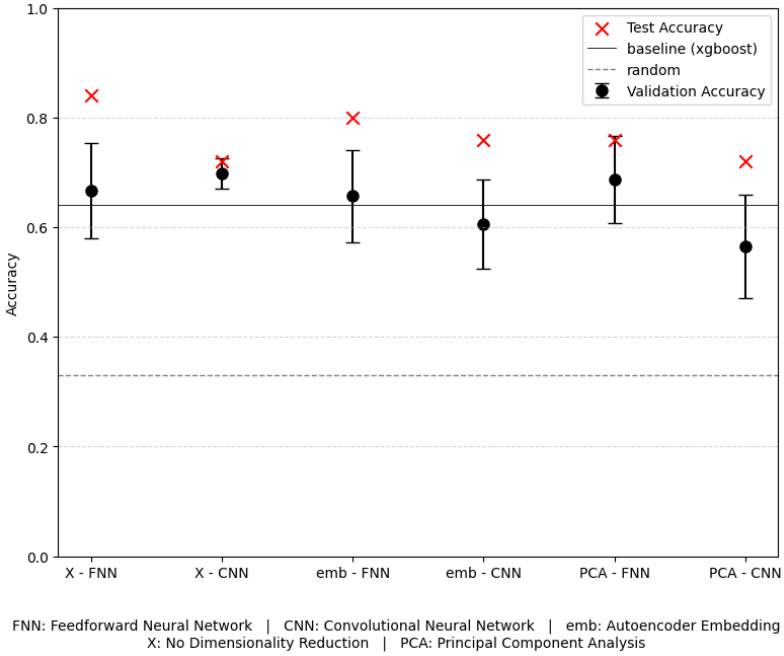


Figure 4.6: 3-class split plot showing classification accuracy for all six DRT–NN configurations (X-axis). The Y-axis displays accuracy scores. Red 'X' markers indicate test set accuracy, and black whiskers represent mean validation accuracy with standard deviation from 5-fold cross-validation. The black horizontal line marks the XGBoost baseline (0.64), and the gray line represents the random classification baseline (0.33). In this setting, Class 0 included the Control group, Class 1 combined Groups 1 and 3, and Class 2 combined Groups 2 and 4. Validation accuracy was more consistent across configurations than in the 5-class scenario, ranging from 0.57 (PCA–CNN) to 0.70 (raw–CNN), with most standard deviations between 0.08 and 0.09, except for raw–CNN (0.03). Test accuracy ranged from 0.72 for CNN and PCA–CNN, to 0.84 for raw–FNN, followed by 0.80 for emb–FNN.

4.3.3 2-class split

A binary (2-class) classification scenario was considered to assess potential differences between the control and experimental setups combined into one group. In this setting, the random baseline is marked by the accuracy of 0.50. The XGBoost approach resulted in the accuracy of 0.84. Mean validation accuracy was consistent across architectures ranging from 0.71 for PCA-CNN to 0.77 for PCA-FNN. Standard deviations varied between 0.04 for emb-CNN and 0.14 for X-CNN. Note, that for the test set, none of the tested models outperformed the XGBoost. However, all configurations based on FNN exactly matched the test accuracy of XGBoost (0.84). A somewhat lower accuracy of 0.80 was calculated for PCA-CNN, while all CNN-based architectures performed the worst. The overall lowest accuracy of 0.62 was obtained for X-CNN followed by emb-CNN at 0.68 (fig. 4.7 and tab. **??**).

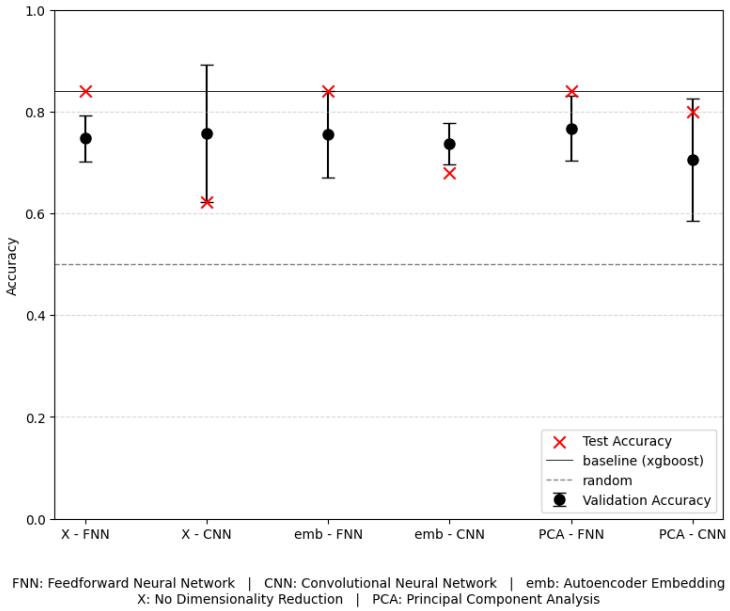


Figure 4.7: 2-class split plot showing classification accuracy for all six DRT–NN configurations (X-axis). The Y-axis displays accuracy scores. Red 'X' markers indicate test set accuracy, and black whiskers represent mean validation accuracy with standard deviation from 5-fold cross-validation. The black horizontal line marks the XGBoost baseline (0.84), and the gray line shows the random classification baseline (0.50). In this binary scenario, the Control group was classified as Class 0, while experimental groups were combined into Class 1. Validation accuracy was relatively consistent across configurations, ranging from 0.71 (PCA–CNN) to 0.77 (PCA–FNN), with standard deviations between 0.04 (emb–CNN) and 0.14 (X–CNN). On the test set, no model surpassed the XGBoost baseline. However, all FNN-based configurations matched the XGBoost accuracy (0.84), while CNN-based models performed worse, with the lowest result observed for X–CNN (0.62), followed by emb–CNN (0.68).

Chapter 5

DISCUSSION AND CONCLUSIONS

5.1 Discussion

5.1.1 Data inspection and preprocessing

Initial data exploration revealed a high proportion of zero abundance cells. Hence, from the statistical perspective, the dataset represents zero-inflated data that leads to the violation of statistical properties underlying standard statistical methodology. Biologically, such sparse structure is characteristic for microbiome studies where zero-abundance either represents taxa really missing in the analysed environment or taxa that are abundant but below the detection sensitivity of the sequencing method. Despite this, several machine learning models achieved strong classification performance, suggesting that zero inflation did not substantially reduce predictive power as stated in Haghani et al. (2017).

From the perspective of the application of DL methods, high variation in abundance across features and samples posed a risk of model training instability and biased learning. To address the problem of zero-inflation, the CLR transformation was applied at data preprocessing step, reducing variance range disparities and helping to ensure that all bacterial families, regardless of abundance, were appropriately represented in the modeling process (Yerke et al., 2024). Usefulness of this method is also proved by its application to datasets which allows usage of traditional statistical methods (den Boogaart & Tolosana-Delgado, 2013; Gloor et al., 2017).

5.1.2 Dimensionality reduction and model performance

Among dimensionality reduction techniques, the autoencoder-based embedding was the most effective. Particularly in the 5-class classification scenario, where it reached a test accuracy of 0.84. This performance was unmatched by any other approach, including the baseline provided by XGBoost. This result suggests that the embedding captured meaningful patterns in the microbial abundance profiles. Across all classification scenarios (5, 3, or 2 classes), neither PCA nor AE-based representations clearly emerged as the best. Concluding, the impact of DRTs appeared subtle and configuration-dependent. No-

tably, convolutional neural network based classifiers consistently underperformed. Often ranking lowest in terms of test accuracy. This suggests that the representations introduced by DRTs may have resulted in representations that were not well-suited for convolutional feature extraction that focuses on spatial-like patterns that were apparently not present in the analyzed data. Lastly, test accuracy often exceeded validation accuracy. This was likely because, during cross-validation, models were trained on only 80% of the validation set and validated on the remaining 20%. In contrast, for the final evaluation on the test set, the models were trained on the entire validation dataset, giving them access to more data for learning and potentially resulting in better performance.

5.1.3 Influence of class complexity

Classification accuracy increased consistently as the number of classes was reduced. This trend is expected, since fewer classes reduce the likelihood of misclassifications that is clearly reflected by increasing accuracy of the random baseline (0.20 for 5 classes, 0.33 for 3, and 0.50 for a binary classification). FNN-based classification appeared to be the most robust across classification scenarios. Interestingly, the most consistent performance, both in terms of accuracy and standard deviation for validation datasets, was observed in the 3-class scenario. This may reflect a biologically meaningful, true class separation, potentially corresponding to specific treatment groups. The control fish had different microbiome patterns compared to the fish that received probiotics. In the supplemented groups, the type of feed seemed to be the main factor influencing microbiome differences.

5.2 Conclusions

5.2.1 Training behavior and generalization

While not included in the main results section, training dynamics observed during cross-validation provided useful insights into model behavior. In setups without dimensionality reduction, Feedforward Neural Networks demonstrated stable and consistent training across folds expressed by smooth decay of loss curves and little to no overfitting. In contrast, CNN-based architectures without DRT tended to overfit, typically around 150 epochs, and exhibited more variable and less smooth loss trajectories.

When DRT methods such as PCA or AE-embeddings were applied, the loss curves for CNNs became smoother, though still somewhat inconsistent across validation runs. Notably, FNNs trained on reduced representations often converged more quickly, typically reaching a plateau in validation performance within the first 50 epochs, suggesting that DRT may enable faster and more efficient learning for simpler architectures (Velliangiri et al., 2019).

The observed ceiling of 0.84 test accuracy, equivalent to correctly classifying 21 out of 25 samples, was not exceeded by any model configuration nor the baseline. This suggests

the presence of potential outlier samples or inherent biological noise within the dataset. Improving performance beyond this level may require models with greater capacity meaning deeper architectures, more parameters which allows models for capturing more complex relationships within the data, or access to additional features, such as environmental variables or host-related metadata. One particularly interesting approach was presented by Nguyen et al. (2017), where taxonomic information was used to construct phylogenetic trees that were visualized as 2D images, with pixel density representing bacterial abundance. These images were then classified using convolutional neural networks. While promising, such an approach would require substantially larger datasets to avoid overfitting. Indeed, in this study, overfitting was frequently observed in CNN-based configurations, likely due to their higher representational capacity. When applied to small datasets, such models are more prone to capturing noise rather than true signal. This underscores the critical need to match model complexity to dataset size and true underlying complexity of the data. In this context, dimensionality reduction techniques serve as a valuable tool for mitigating overfitting by refining the input space. By reducing input dimensionality while retaining essential information, DRTs may help more complex models generalize effectively, even when data is small (Rajput et al., 2023).

5.2.2 Computational efficiency and scalability

Although computational resource usage was not quantitatively assessed in this study, observations indicated that simpler architectures, particularly feedforward neural networks, required much less computational time compared to more complex models. Among the dimensionality reduction techniques, PCA was considerably faster to compute than AE-embeddings, which involved training the whole autoencoder network. Nevertheless, all computations were completed within 15 minutes per configuration on a standard home-grade PC, suggesting that, for this dataset size and model complexity, resource demands were minimal - with RAM usage peaking at 2.5 GB and CPU utilization at 37%. However, it should be noted that this efficiency may not scale well. Larger datasets or higher-capacity models, such as deeper convolutional architectures or more complex AE, are likely to require substantially more computational resources that may go beyond the above mentioned hardware architecture. As such, computational scalability remains an important consideration for future studies aiming to apply similar methods to larger or more diverse microbiome datasets.

5.2.3 DRT selection

At the initial stage of the study, other dimensionality reduction techniques such as t-SNE and UMAP were also considered (Medina et al., 2022). However, they were excluded due to their poorer convergence, reduced robustness, and sensitivity to hyperparameter tuning. Given the need to simultaneously evaluate both dimensionality reduction techniques and deep learning architectures, it was important to use a robust approach. This

allowed the focus to remain on assessing the performance of the neural network models without introducing additional variability from unstable dimensionality reduction methods (George et al., 2021; Kostic et al., 2015; Xu et al., 2020).

5.3 Limitations and future research

5.3.1 Limitations

While this study yielded promising results, several limitations should be addressed. First the analysis did not include an assessment of feature importance, which limits the biological interpretability of the models and thus it remains unclear which bacterial families contributed most significantly to classification performance. Second, the classifier models used were limited to only two architectures (FNN, CNN) constraining scope of model comparison. Other potentially relevant architectures like the suggested by X. Chen et al. (2022) and Maryam et al. (2024) Graph and Recurrent neural networks as well as transformer architectures were not explored (Przymus et al., 2025). Another limitation of this study is as explained above, relatively small dataset size, which constrains generalizability of the results and limits the ability of high capacity models to learn effectively without overfitting. Finally, the dataset lacked additional explanatory variables, such as host metadata, environmental parameters, or feed composition details, which could have provided valuable context for interpreting microbial variation and lack of those can limit the model's ability to account for confounding factors which may bias the results, as stated by Vandenbroucke et al. (2007). The absence of such features limits the potential for deeper biological inference.

5.3.2 Future research

Building upon the limitations identified in this study, several key directions for future research emerge. Following the promising result achieved with the autoencoder-based embedding, where a test accuracy of 0.84 was reached in the 5-class classification scenario, there is a clear opportunity to enhance both the interpretability and robustness of the models. Most notably, assessing feature importance would help identify the specific bacterial families most influential in classification, opening the door to biological interpretation, particularly in the context of probiotic supplementation strategies. Furthermore, incorporating additional explanatory variables, such as host-related or environmental metadata, could improve model performance and reduce noise-driven variance. Statistically, the use of simulated datasets should be explored to evaluate the properties of the applied dimensionality reduction techniques and classifiers, especially in low-sample, high-dimensional settings. Expanding the range of model architectures and increasing dataset size will also be critical in advancing the reliability and generalizability of future findings.

Data and code availability

All the data and code used in this study is available on git repository under link: https://github.com/paq88/Deep-Learning-in-classification-of-common-Carp-intestine-classification

BIBLIOGRAPHY

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283.

Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *44*, 139–160. https://doi.org/10.1111/j.2517-6161.1982.tb01195.x

Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, *13*, 27–31. https://doi.org/10.1109/45.329294

Behera, S., Catreux, S., Rossi, M., Truong, S., Huang, Z., Ruehle, M., Visvanath, A., Parnaby, G., Roddey, C., Onuchic, V., Finocchio, A., Cameron, D. L., English, A., Mehtalia, S., Han, J., Mehio, R., & Sedlazeck, F. J. (2024). Comprehensive genome analysis and variant detection at scale using dragen. *Nature Biotechnology*. https://doi.org/10.1038/s41587-024-02382-1

Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review*, *57*, 28. https://doi.org/10.1007/s10462-023-10662-6

Blaser, M. J. (2014). The microbiome revolution. *The Journal of Clinical Investigation*, *124*, 4162–4165. https://doi.org/10.1172/JCI78366

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C. C., Al-Ghalith, G. A., Alexander, H., Alm, E. J., Arumugam, M., Asnicar, F., Bai, Y., Bisanz, J. E., Bittinger, K., Brejnrod, A., Brislawn, C. J., Brown, C. T., Callahan, B. J., Caraballo-Rodríguez, A. M., Chase, J., … Caporaso, J. G. (2019). Reproducible, interactive, scalable and extensible microbiome data science using qiime 2. *Nature Biotechnology*, *37*, 852–857. https://doi.org/10.1038/s41587-019-0209-9

Chen, T., & Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, X., Zhu, Z., Zhang, W., Wang, Y., Wang, F., Yang, J., & Wong, K.-C. (2022). Human disease prediction from microbiome data by multiple feature fusion and deep learning. *iScience*, *25*, 104081. https://doi.org/10.1016/j.isci.2022.104081

Chollet, F., et al. (2015). Keras.

Crall, J. D., Gravish, N., Mountcastle, A. M., & Combes, S. A. (2015). Beetag: A low-cost, image-based tracking system for the study of animal behavior and locomotion. *PLOS ONE*, *10*, e0136487. https://doi.org/10.1371/journal.pone.0136487

den Boogaart, K. G., & Tolosana-Delgado, R. (2013). *Analyzing compositional data with r* (Vol. 122). Springer.

der Auwera, G. A., & O'Connor, B. D. (2020). *Genomics in the cloud: Using docker, gatk, and wdl in terra*. O'Reilly Media.

Durán, C., Ciucci, S., Palladini, A., Ijaz, U. Z., Zippo, A. G., Sterbini, F. P., Masucci, L., Cammarota, G., Ianiro, G., Spuul, P., Schroeder, M., Grill, S. W., Parsons, B. N., Pritchard, D. M., Posteraro, B., Sanguinetti, M., Gasbarrini, G., Gasbarrini, A., & Cannistraci, C. V. (2021). Nonlinear machine learning pattern recognition and bacteria-metabolite multilayer network analysis of perturbed gastric microbiome. *Nature Communications*, *12*, 1926. https://doi.org/10.1038/s41467-021-22135-x

Feng, C. X. (2021). A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. *Journal of Statistical Distributions and Applications*, *8*, 8. https://doi.org/10.1186/s40488-021-00121-4

Fukushima, K. (1979). Self-organization of a neural network which gives position-invariant response. *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*, 291–293.

George, A., Cameron, M., Gibraan, R., Antonio, G., Yoshiki, V.-B., Gal, M., & Rob, K. (2021). Uniform manifold approximation and projection (umap) reveals composite patterns and resolves visualization artifacts in microbiome data [doi: 10.1128/msystems.00691-21]. *mSystems*, *6*, 10.1128/msystems.00691–21. https://doi.org/10.1128/msystems.00691-21

Gilbert, J. A., Blaser, M. J., Caporaso, J. G., Jansson, J. K., Lynch, S. V., & Knight, R. (2018). Current understanding of the human microbiome. *Nature Medicine*, *24*, 392–400. https://doi.org/10.1038/nm.4517

Gloor, G. B., Macklaim, J. M., Pawlowsky-Glahn, V., & Egozcue, J. J. (2017). Microbiome datasets are compositional: And this is not optional. *Frontiers in Microbiology*, *8*. https://doi.org/10.3389/fmicb.2017.02224

Gloor, G. B., Wu, J. R., Pawlowsky-Glahn, V., & Egozcue, J. J. (2016). It's all relative: Analyzing microbiome data as compositions. *Annals of Epidemiology*, *26*, 322–329. https://doi.org/https://doi.org/10.1016/j.annepidem.2016.03.003

GNU, P. (2007). Free software foundation. bash (3.2. 48)[unix shell program].

Haghani, S., Sedehi, M., & Kheiri, S. (2017). Artificial neural network to modeling zero-inflated count data: Application to predicting number of return to blood donation. *Journal of Research in Health Sciences*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with numpy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, *9*, 90–95.

Johnson, J. S., Spakowicz, D. J., Hong, B.-Y., Petersen, L. M., Demkowicz, P., Chen, L., Leopold, S. R., Hanson, B. M., Agresta, H. O., Gerstein, M., Sodergren, E., & Weinstock, G. M. (2019). Evaluation of 16s rrna gene sequencing for species and strain-level microbiome analysis. *Nature Communications*, *10*, 5029. https://doi.org/10.1038/s41467-019-13036-1

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, *596*, 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kanika, N. H., Liaqat, N., Chen, H., Ke, J., Lu, G., Wang, J., & Wang, C. (2025). Fish gut microbiome and its application in aquaculture and biological conservation. *Frontiers in Microbiology*, *15*. https://doi.org/10.3389/fmicb.2024.1521048

Karashchuk, P., Tuthill, J. C., & Brunton, B. W. (2021). The dannce of the rats: A new toolkit for 3d tracking of animal behavior. *Nature Methods*, *18*, 460–462. https://doi.org/10.1038/s41592-021-01110-w

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398. https://doi.org/https://doi.org/10.1016/j.ymssp.2020.107398

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90).

Koonce, B. (2021). Resnet 50. In B. Koonce (Ed.), *Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization* (pp. 63–72). Apress. https://doi.org/10.1007/978-1-4842-6168-2_6

Kostic, A. D., Gevers, D., Siljander, H., Vatanen, T., Hyötyläinen, T., Hämäläinen, A.-M., Peet, A., Tillmann, V., Pöhö, P., Mattila, I., Lähdesmäki, H., Franzosa, E. A., Vaarala, O., de Goffau, M., Harmsen, H., Ilonen, J., Virtanen, S. M., Clish, C. B., Orešič, M., … Xavier, R. J. (2015). The dynamics of the human infant gut microbiome in development and in progression toward type 1 diabetes [doi: 10.1016/j.chom.2015.01.001]. *Cell Host & Microbe*, *17*, 260–273. https://doi.org/10.1016/j.chom.2015.01.001

Kristensen, N. B., Bryrup, T., Allin, K. H., Nielsen, T., Hansen, T. H., & Pedersen, O. (2016). Alterations in fecal microbiota composition by probiotic supplementation in healthy adults: A systematic review of randomized controlled trials. *Genome Medicine*, *8*, 52. https://doi.org/10.1186/s13073-016-0300-5

Lee, B.-C., Hung, C.-W., Lin, C.-Y., Shih, C.-H., & Tsai, H.-J. (2019). Oral administration of transgenic biosafe microorganism containing antimicrobial peptide enhances the survival of tilapia fry infected bacterial pathogen. *Fish & Shellfish Immunology*, *95*, 606–616. https://doi.org/https://doi.org/10.1016/j.fsi.2019.10.052

Lin, H., & Peddada, S. D. (2020). Analysis of microbial compositions: A review of normalization and differential abundance analysis. *npj Biofilms and Microbiomes*, *6*, 60. https://doi.org/10.1038/s41522-020-00160-w

Liou, C.-Y., Cheng, W.-C., Liou, J.-W., & Liou, D.-R. (2014). Autoencoder for words. *Neurocomputing*, *139*, 84–96. https://doi.org/https://doi.org/10.1016/j.neucom.2013.09.055

Lloyd-Price, J., Abu-Ali, G., & Huttenhower, C. (2016). The healthy human microbiome. *Genome Medicine*, *8*, 51. https://doi.org/10.1186/s13073-016-0307-y

Maryam, Rehman, M. U., Hussain, I., Tayara, H., & Chong, K. T. (2024). A graph neural network approach for predicting drug susceptibility in the human microbiome. *Computers in Biology and Medicine*, *179*, 108729. https://doi.org/https://doi.org/10.1016/j.compbiomed.2024.108729

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259

McKinney, W., et al. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56.

Medina, R. H., Kutuzova, S., Nielsen, K. N., Johansen, J., Hansen, L. H., Nielsen, M., & Rasmussen, S. (2022). Machine learning and deep learning applications in microbiome research. *ISME Communications*, *2*, 98. https://doi.org/10.1038/s43705-022-00182-9

Modi, A., Vai, S., Caramelli, D., & Lari, M. (2021). The illumina sequencing protocol and the novaseq 6000 system. https://doi.org/10.1007/978-1-0716-1099-2_2

Nayak, S. (2010). Probiotics and immunity: A fish perspective. *Fish & Shellfish Immunology*, *29*, 2–14. https://doi.org/10.1016/j.fsi.2010.02.017

Nguyen, T. H., Chevaleyre, Y., Prifti, E., Sokolovska, N., & Zucker, J.-D. (2017). Deep learning for metagenomic data: Using 2d embeddings and convolutional neural networks.

OpenAI. (2023). Chatgpt.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*, 559–572. https://doi.org/10.1080/14786440109462720

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*, 2825–2830.

Przymus, P., Rykaczewski, K., Martín-Segura, A., Truu, J., Pau, E. C. D. S., Kolev, M., Naskinova, I., Gruca, A., Sampri, A., Frohme, M., & Nechyporenko, A. (2025). Deep learning in microbiome analysis: A comprehensive review of neural network models. *Frontiers in Microbiology*, *15*. https://doi.org/10.3389/fmicb.2024.1516667

Rajput, D., Wang, W.-J., & Chen, C.-C. (2023). Evaluation of a decided sample size in machine learning applications. *BMC Bioinformatics*, *24*, 48. https://doi.org/10.1186/s12859-023-05156-9

Rossum, G. V., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. https://doi.org/10.1038/323533a0

Sehnal, L., Brammer-Robbins, E., Wormington, A. M., Blaha, L., Bisesi, J., Larkin, I., Martyniuk, C. J., Simonin, M., & Adamovsky, O. (2021). Microbiome composition and function in aquatic vertebrates: Small organisms making big impacts on aquatic animal health. *Frontiers in Microbiology*, *Volume 12 - 2021*.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., & Vinyals. (2025). Gemini: A family of highly capable multimodal models.

Vandenbroucke, J. P., von Elm, E., Altman, D. G., Gøtzsche, P. C., Mulrow, C. D., Pocock, S. J., Poole, C., Schlesselman, J. J., & Egger, M. (2007). Strengthening the reporting of observational studies in epidemiology (strobe): Explanation and elaboration. *PLoS Medicine*, *4*, e297. https://doi.org/10.1371/journal.pmed.0040297

Velliangiri, S., Alagumuthukrishnan, S., & joseph, S. I. T. (2019). A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science*, *165*, 104–111. https://doi.org/https://doi.org/10.1016/j.procs.2020.01.079

Wu, J., Li, Y., & Ma, Y. (2021). Comparison of xgboost and the neural network model on the class-balanced datasets. *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 457–461. https://doi.org/10.1109/ICFTIC54370.2021.9647373

Xu, X., Xie, Z., Yang, Z., Li, D., & Xu, X. (2020). A t-sne based classification approach to compositional microbiome data. *Frontiers in Genetics*, *11*. https://doi.org/10.3389/fgene.2020.620143

Yerke, A., Brumit, D. F., & Fodor, A. A. (2024). Proportion-based normalizations outperform compositional data transformations in machine learning applications. *Microbiome*, *12*, 45. https://doi.org/10.1186/s40168-023-01747-z

Zheng, J., Wittouck, S., Salvetti, E., Franz, C. M., Harris, H. M., Mattarelli, P., O'Toole, P. W., Pot, B., Vandamme, P., Walter, J., Watanabe, K., Wuyts, S., Felis, G. E., Gänzle, M. G., & Lebeer, S. (2020). A taxonomic note on the genus lactobacillus: Description of 23 novel genera, emended description of the genus lactobacillus beijerinck 1901, and union of lactobacillaceae and leuconostocaceae. *International Journal of Systematic and Evolutionary Microbiology*, *70*, 2782–2858. https://doi.org/10.1099/ijsem.0.004107

Chapter 6

APPENDIX

Table 6.1: Validation accuracy (mean ± SD) and test accuracy across all classification scenarios.

Configuration	**5-Class**	**3-Class**	**2-Class**
raw-FNN	0.58 (±0.06) / 0.68	0.67 (±0.09) / 0.84	0.75 (±0.05) / 0.84
raw-CNN	0.43 (±0.12) / 0.36	0.70 (±0.03) / 0.72	0.76 (±0.14) / 0.62
emb-FNN	0.53 (±0.12) / 0.84	0.66 (±0.08) / 0.80	0.76 (±0.09) / 0.84
emb-CNN	0.47 (±0.12) / 0.52	0.61 (±0.08) / 0.76	0.74 (±0.04) / 0.68
PCA-FNN	0.60 (±0.10) / 0.60	0.69 (±0.08) / 0.76	0.77 (±0.06) / 0.84
PCA-CNN	0.32 (±0.05) / 0.48	0.57 (±0.09) / 0.72	0.71 (±0.12) / 0.80