Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

Multiclass Disease Classification via Microbiome Profiling with Transformers

Mohammed Khalfan MKHALFAN@NYU.EDU
New York University
New York, NY 10003, USA

Ching-Tsung(Deron) Tsai CT2840@NYU.EDU
New York University
New York, NY 10003, USA

Editor: Leslie Pack Kaelbling

Abstract

In this study, we develop and assess machine learning models for multi-class classification of diseases using microbiome data, specifically focusing on liver cirrhosis, type II diabetes, obesity, and irritable bowel syndrome. Previous studies used binary classifiers to analyze diseases independently, while our approach employs single multi-class classifiers, which account for the relationships between different disease classes. We compare our models to existing methodologies, such as PopPhy-CNN, Met2Img, and RegMIL, which were previously evaluated in the MetaPheno study. Using evaluation metrics like balanced accuracy, macro F1-Score, and multilabel AUC, we find that our models yield promising results. Our findings emphasize the importance of metadata for effective analysis and interpretation of microbiome data. Future research should explore feature extraction methods, feature selection strategies, and the identification of key species for disease classification to improve model performance.

Keywords: Microbiome Profiling, Disease Classification, Transformers

1. Introduction

Research has demonstrated that dysbiosis, or an imbalanced intestinal microbiota, plays a factor in many human disorders (Lynch and Pedersen, 2016). Systematic characterization of this microbiome offers the chance to develop non-invasive methods for diagnosing major diseases. This is an active area of research which is being driven by advances in the field of machine learning and deep learning (Su et al., 2022; Khan and Kelly, 2019; LaPierre et al., 2019).

In this work, we aimed to develop a transformer-based multi-class diagnostic deep learning model that can predict various diseases based on microbiome profiling. While transformers were originally designed for processing sequential data, their self-attention mechanism is also well-suited for capturing long-range dependencies in non-sequential data. Recently, transformers have been applied to fields such as genomics, where they have shown promise for analyzing non-sequential quantitative data (Clauwaert et al., 2021; Jubair et al., 2021; Dalla-Torre et al., 2023). In our study, we aimed to leverage the strengths of transformers for disease prediction based on microbiome profiling, where the relative abundance of mi-

crobial taxa serves as the input data. We hypothesized that the self-attention mechanism of transformers could enable our model to capture complex relationships between microbial taxa and their potential associations with disease outcomes.

In the MetAML study, the authors utilized four traditional machine learning classifiers for metagenomic-based prediction tasks, specifically SVM, RF, Lasso, and ENet. Our objective was to surpass the performance achieved in their study by implementing transformer-based deep learning methods. Moreover, since the MetaPheno (LaPierre et al., 2019) study compares numerous newer deep learning algorithms to the MetAML study, we also compare our results with theirs to gain further insights into our model's performance. To evaluate performance, we employ the same evaluation metrics used in the MetaPheno study, including overall accuracy, F1 score, and area under the curve. By using these metrics, we can compare our results with the published work and determine the effectiveness of our approach.

In addition, we studied the impact of metadata, such as gender and country, on disease prediction accuracy. Finally, we tested gradient boosted decision trees (Friedman, 2000), given that tabular data classification problems are still dominated by this method due to their short training time and robustness. By conducting these additional experiments, we were able to gain a deeper understanding of the strengths and limitations of our models and further validate their effectiveness.

2. Data

We used the dataset from MetAML (Pasolli et al., 2016), which is publicity available on github. The dataset contains 3610 samples with 3513 features each, including over 200 patient metadata and 3302 microbiome expression data. The estimated relative abundance of microbial taxa has already been quantified. The dataset includes 20 classes of diseases, but many of them have limited counts. Therefore, we will focus on the top 6 outcome classes that include 3271 patients.

The estimated relative abundance of microbial taxa is already quantified, which means that feature extraction has already been performed. We retained only the species-level data and set a threshold for the features to ensure that only relevant and informative features were included in the models. Specifically, we relied on the threshold criteria specified in Su et al. (2022). The thresholds for the features are existence in more than 5% of the subjects with more than 0.15% relative abundance. For the metadata features, we removed features with more than 20% of missingness, and we conducted a single KNN imputation to fill in the missing values. The remaining data contained 2811 samples with 332 features, including 3 metadata (bodysite, gender, country) and 328 microbiome profiles. Finally, we performed a random train-test split on the data, where we used 70% of the data for training the models and 30% for evaluation.

3. Methods

We employed two separate transformer based methods in this project, and also tested gradient boosted decision trees.

3.1 TabPFN

TabPFN is a Prior-Data Fitted Network (PFN) designed as a trained Transformer capable of performing supervised classification on small tabular datasets in under a second (Hollmann et al., 2022). This method requires no hyperparameter tuning and claims to compete with state-of-the-art classification techniques. TabPFN is trained offline once using synthetic datasets. Specifically, the authors trained a 12-layer transformer for 18,000 batches, with each batch containing 512 synthetically generated datasets. This offline training phase only needs to be executed once. During this process, the transformer encodes each feature vector and label as a token, enabling token representations to attend to one another.

For inference, the pre-trained model is applied to unseen real-world datasets. The model takes the set of training sample $D_{train} := (x_1, y_1), ..., (x_n, y_n)$ and X_{test} as input and produces the Posterior Predictive Distribution (PPD) of y, given X_{test} and D_{train}, in a single forward pass. The PPD class probabilities are then utilized as predictions. This Bayesian statistical method employs the posterior predictive distribution as a means of estimating the probability distribution of a new data point, considering both the observed data and a prior distribution of the model parameters.

3.1.1 Caveats and Experiments

While the authors hailed TabPFN as a revolutionary and radical approach to tabular classification, it is not without its limitations.

One notable caveat is the error raised when attempting to use a training set larger than 1024, which can be manually overridden by employing the *overwrite_warning* parameter. In order to examine the application of TabPFN within its intended design, we explored various strategies to address the 1024 training size constraint. One approach involved dividing the training data into two distinct sets with maximally dissimilar samples and generating predictions for each set, but this led to inconsistent predictions for some test set samples. An alternative method drew inspiration from machine learning, using a bootstrap aggregation ensemble (bagging) technique in which the model was executed 100 times, each time selecting 1000 samples randomly, with replacement, from the train set. Final predictions were determined using a majority rules voting method. Lastly, we set the *overwrite_warning* parameter to True and provided the entire train set. As is often the case with conventional deep learning techniques, this approach of using a larger training set yielded the best performance in comparison to the other two methods designed to adhere to TabPFN's recommended constraints.

Another limitation pertained to the maximum number of features allowed per sample, which is restricted to 100 without an option to bypass this constraint. Consequently, we had to engage in feature selection. We examined various strategies in this regard, such as selecting the top 97 expression columns by sum (with 3 columns reserved for metadata), and choosing the top 97 expression columns by variance, under the premise that extreme differences in microbial abundance, rather than merely high prevalence of certain microbes, could be significant for disease characterization. Lastly, inspired by the bagging approach used to address the training size limitation, and to sidestep manual feature selection, we implemented a method similar to bagging in which the model was executed 100 times,

each time randomly selecting 97 columns without replacement. Intriguingly, this approach outperformed all other feature selection methods.

3.1.2 Assessing the importance of Metadata

In an initial effort to identify the features that may be important for the model's performance, we repeated the best-performing method of randomly selecting 97 columns and running the model 100 times. However, this time we selected 100 columns exclusively from the expression data, entirely excluding the metadata. Notably, eliminating the metadata led to the most significant decline in performance across all experimental conditions.

3.1.3 Hyperparameters

Although TabPFN does not require tuning traditional deep learning hyperparameters such as learning rate and batch size, the model does include adjustable parameters.

The $n\_ensemble\_configurations$ parameter facilitates ensemble learning by default, controlling the number of model predictions combined to enhance accuracy and reduce overfitting. This approach involves creating distinct variations of the same model by rotating features and classes, thereby contributing to improved model performance. Despite the documentation suggesting that accuracy increases with this parameter, we did not observe this trend.

By default, the $no\_preprocess\_mode$ parameter is set to False. The authors advise against preprocessing inputs for TabPFN, as it internally applies z-score normalization per feature (fitted on the training set) and log-scales outliers heuristically, and state that preprocessing is crucial for ensuring that the real-world dataset aligns with the distribution of synthetic datasets encountered during training. Setting this parameter to True disables TabPFN's internal preprocessing. We evaluated this parameter across a range of $n\_ensemble\_configurations$ and found that the default setting consistently yielded higher performance.

We examined $multiclass_decoder$ and $feature_shift_decoder$, which allow random shifts in classes and features, respectively, for each ensemble configuration. After testing with various $n_ensemble_configurations$, we found that the default settings resulted in a higher macro F1 in all cases. Therefore, we recommend using the default settings for these parameters.

3.2 FT-Transformer

The Feature Tokenizer Transformer (FT-Transformer) (Gorishniy et al., 2021), is an innovative model that leverages Feature Tokenizer and Transformer techniques. It offers a flexible approach to handle both numeric and categorical features by transforming them into embeddings and stacking them as the same input for Transformer blocks. For categorical features, it first converts them into one-hot encoding representation. Afterward, all features undergo an element-wise multiplication to produce a vector with dimension k*d, where k represents the number of features after one-hot encoding and d is the dimension of vector representation. A feature-level bias is then added to the encoding. Next, a CLS token is appended to the stacked sequence, and L transformer layers are applied to the data. The final outcome is predicted based on the presentation of the CLS token. Once the transformer blocks have finished, the output y is normalized using layer normalization,

and then it goes through a ReLU activation, and a linear layer with dropout. Overall, FT-Transformer provides a powerful and flexible way to embed different types of features and has shown promising results in various deep learning tasks.

3.2.1 Hyperparameters

In this study, a 3-fold stratified cross-validation was performed on the training data, using the cross-entropy loss function and Adam optimizer with weight decay. The hyperparameters that were tuned included the learning rate, weight decay, batch size, number of transformer blocks, and dropout rate. Ray Tune was used for hyperparameter tuning with 3 rounds of 200 random trials. Random search was performed with an adjusted parameter space based on the results of previous rounds. After the hyperparameter tuning, the final parameters were determined as follows: learning rate = 5e-5, batch size = 128, weight decay = 1e-2, number of transformer blocks = 3, and dropout rate after each block = 0.2. The results showed that learning rate and batch size had the most impact on the performance. The suggested value from the author or the default value in PyTorch was used for weight decay and the number of transformer blocks since they showed negligible impact.

3.3 Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (XGBoost) is an ensemble learning method that builds a series of decision trees to make predictions by minimizing the loss function. We tested the XGBoost algorithm using the default parameters, which includes a maximum tree depth of 6, a learning rate of 0.3, and a subsample ratio of 1.0.

4. Results

In our study, we assessed the performance of the models we developed against several methodologies, including PopPhy-CNN (Reiman et al., 2018), Met2Img (Nguyen et al., 2018), and RegMIL (Rahman and Rangwala, 2018), which were previously evaluated in the MetaPheno study against the MetAML dataset and methods. The diseases that we considered in our analysis were liver cirrhosis, type II diabetes, obesity, and irritable bowel syndrome, which were the same diseases used in the MetaPheno study. It is important to note that previous methods analyzed these diseases independently using a binary classifier, whereas the models we have developed are single multi-class classifiers, representing a significant departure from prior approaches. Our results are presented within this context.

We employed several evaluation metrics in our analysis to measure the performance of the models, including accuracy, balanced accuracy, macro F1-Score, and multilabel AUC for our highly imbalanced multi-class classification task. We used the F1-Score, which is the macro F1-Score, which is the arithmetic mean of the F1-Scores for each class, as our main criteria.

The optimized FTT and TabPFN models showed a moderate improvement of 6% and 4%, respectively, compared to their default counterparts. However, the XGBoost model outperformed all other models across all criteria with a BAC of 0.7364 and macro-F1 of 0.7331, followed by the FTT and TabPFN models.

Table 1: Performance Metrics

Model	**BAC**	**macro-F1**	**Accuracy**
FTT-optimized	0.7346	0.6418	0.7853
FTT-default	0.6746	0.6075	0.7794
TabPFN-optimized	0.5550	0.6001	0.8493
TabPFN-default	0.5130	0.5717	0.8291
XGBoost	0.7364	0.7331	0.8801

5. Discussion

Compared to previous studies, our approach to building and testing multiclass classifiers uses a different methodology, which involves training models that consider the relationships between the different disease classes. In contrast, using binary classifiers for multiclass disease classification from microbiome data can result in statistical inefficiencies due to the lack of explicit consideration of class interactions. Our classifier demonstrated promising results with multilabel AUC of 0.95 for FTT amd AUC of 0.91 for classifying obesity in TabPFN. In comparison, traditional machine learning classifiers reported an average AUC of only 0.68 in classifying obesity in LaPierre et al. (2019). However, it is important to note that our data originally came from binary studies, and our model assumes that 'no disease' indicates the absence of all considered diseases. In contrast, the original studies only indicated the absence of the specific disease being tested in the binary case.

A critical stage in developing machine learning models for metagenomic data is feature extraction, which involves transforming raw sequence reads into structured data. Although our project did not explore different methods of feature extraction due to time constraints, it would be interesting to investigate their impact on model performance, and allow us to use raw sequence data from other studies where extracted feature data is unavailable.

As the number of allowed features for TabPFN is limited, we had an opportunity to explore feature selection, and found that using random features yielded better performance compared to manual feature selection. Previous studies have achieved optimal performance using fewer than 70 species (LaPierre et al., 2019). Thus, objectives for future research could include identifying key species that are most effective in discriminating between disease classes.

Finally, our findings suggest that metadata is a crucial component for the effective analysis of microbiome data. However, an updated version of the dataset, which includes additional data and is now available as an R package, has omitted the metadata columns. We caution against excluding metadata, as it provides important context and enables the interpretation of the results obtained from the analysis. Therefore, we recommend that future studies continue to include metadata in their analyses.

6. Team Work

Mohammed Khalfan	**Deron**
Project conception	Data pre-processing
Literature search and review	Method 2: FT-Transfomer
Getting data	Consulting for Method 1
Method 1: TabPFN	Results analysis
Method 3: Gradient boosted trees	Writing
Writing	

References

Jim Clauwaert, Gerben Menschaert, and Willem Waegeman. Explainability in transformer models for functional genomics. *Briefings in Bioinformatics*, 22(5), 04 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab060. URL https://doi.org/10.1093/bib/bbab060. bbab060.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, 2023. doi: 10.1101/2023.01.11.523679. URL https://www.biorxiv.org/content/early/2023/01/15/2023.01.11.523679.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second, 2022.

Sheikh Jubair, James R. Tucker, Nathan Henderson, Colin W. Hiebert, Ana Badea, Michael Domaratzki, and W. G. Dilantha Fernando. Gptransformer: A transformer-based deep learning method for predicting fusarium related traits in barley. *Frontiers in Plant Science*, 12, 2021. ISSN 1664-462X. doi: 10.3389/fpls.2021.761402. URL https://www.frontiersin.org/articles/10.3389/fpls.2021.761402.

Saad Khan and Libusha Kelly. Multiclass disease classification from microbial whole-community metagenomes. In *Biocomputing 2020*. World Scientific, 2019. doi: 10.1142/9789811215636_0006. URL https://doi.org/10.1142/9789811215636_0006.

Nathan LaPierre, Christina J Ju, Guangyu Zhou, and Wei Wang. Metapheno: A critical evaluation of deep learning and machine learning in metagenome-based disease prediction. *Methods*, 166:74–82, 2019. doi: 10.1016/j.ymeth.2019.03.003.

Susan V. Lynch and Oluf Pedersen. The human intestinal microbiome in health and disease. *New England Journal of Medicine*, 2016.

Trung Hieu Nguyen, Edi Prifti, Yann Chevaleyre, Nataliya Sokolovska, and Jean-Daniel Zucker. Disease classification in metagenomics with 2d embeddings and deep learning. *arXiv preprint arXiv:1806.09046*, 2018.

Edoardo Pasolli, Duy Tin Truong, Farhan Malik, Levi Waldron, and Nicola Segata. Machine learning meta-analysis of large metagenomic datasets: Tools and biological insights. *PLoS computational biology*, 12(7):e1004977, 2016. doi: 10.1371/journal.pcbi.1004977.

Md Asifur Rahman and Huzefa Rangwala. Regmil: Phenotype classification from metagenomic data. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 145–154. ACM, 2018.

David Reiman, Ahmed A Metwally, and Yang Dai. Popphy-cnn: A phylogenetic tree embedded architecture for convolution neural networks for metagenomic data. *bioRxiv*, page 257931, 2018.

Qi Su, Qin Liu, Raphaela Iris Lau, Jingwan Zhang, Zhilu Xu, Yun Kit Yeoh, Thomas WH Leung, Whitney Tang, Lin Zhang, Jessie QY Liang, et al. Faecal microbiome-based machine learning for multi-class disease diagnosis. *Nature Communications*, 13(1):6818, 2022.