Benchmarking Tabular Representation Models in Transfer Learning Settings

Qixuan Jin
Department of Electrical Engineering and Computer Science, MIT
qixuanj@mit.edu

Talip Uçar
Centre for AI, BioPharmaceuticals R&D, AstraZeneca
talip.ucar@astrazeneca.com

Abstract

Deep learning has revolutionized the transfer of knowledge between similar tasks in data modalities such as images, text, and graphs. However, the same level of success has not been attained in for tabular data. This disparity can be attributed to the inherent absence of structural characteristics, such as spatial and temporal correlations, within common tabular datasets. Moreover, classic methods such as logistic regression and decision trees have been shown to perform competitively with deep learning methods. In this work, we benchmark the classic and deep learning methods specifically within the setting of transfer learning. We offer new benchmarking results for the EHR phenotyping task in the MetaMIMIC dataset and propose a new transfer learning setting of transferring mortality prediction from common to rare cancers with The Cancer Genome Atlas (TCGA).

1 Introduction

In the last decade, deep neural networks have emerged as a powerful tool for learning representations from high-dimensional data. In contrast to classic machine learning methods, neural networks are capable of learning representations from different data sources for a specific task and utilizing that knowledge under new tasks. This has led to significant advancements in the areas of computer vision [1–3] and natural language processing [4, 5], where pretraining and transfer learning have been instrumental in achieving state-of-the-art results [6]. However, the success of these methods has not yet been replicated for tabular data [7, 8], a commonly used data modality in many fields such as healthcare, advertisement, finance, and law. One of the main challenges in applying deep learning to tabular data is the lack of spatial, semantic, or temporal structure in the data that can be leveraged through data augmentation, pretext task generation, and architectural choices. Furthermore, many data augmentation methods, such as cropping, rotation, and color transformation are not suitable for tabular settings. Despite these challenges, there has been a recent increase in interest in developing new methods for representation learning in tabular data [9–16]. In the context of transfer learning, there are multiple promising avenues to explore; i) Data augmentation methods to learn representations from a subset of features [9], such that we can leverage common features across different datasets for transfer learning, ii) Designing new architectures to exploit any inductive biases in a particular tabular data, and iii) Framing the transfer learning setting in a way that it is amenable to the existing methods [17]. In this work, we focus on (iii) and conduct extensive experiments to compare the performance of a diverse set of models for a transfer learning setting defined in [17]. Furthermore, we propose a new transfer learning setting to leverage the representations learned from common cancer types to make predictions for the rare ones in TCGA dataset.

Table Representation Learning Workshop at NeurIPS 2023

2 Related Works

Tabular Representation Learning Representation learning methods for tabular data can be generally categorized into classic and deep learning baselines [18]. Despite the focus on deep learning methods in recent years, classic machine learning methods such as logistic regression and ensemble methods such as decision trees still perform competitively or better than deep learning models, specifically in terms of generalization to novel datasets [19, 7, 8, 20]. Furthermore, the body of literature on deep learning methods is extensive, encompassing a wide range of techniques, including autoencoders [21], their probabilistic variants [22, 23], self-supervised methods [9–11, 24], the differentiable counterparts of classic ensemble methods [11, 14], and the methods based on attention mechanisms and transformers [4, 12, 13, 15, 16, 25–27].

Transfer Learning Transfer learning is a specific paradigm of machine learning in which the useful patterns that a model learns in the source domain are utilized for a similar task in the target domain [28]. Transfer learning can be further categorized by the availability of labeled data in the source and target domains [29], alignment between the input feature spaces [18], and the type of transfer technique [18, 6]. We can categorize some of the recent proposals for learning transferable features for tabular data under two groups: i) Pre-training models across multiple tables with different features [30, 31], ii) Training models on the same set of features to transfer knowledge between different tasks [17]. In this work, we are interested in the data setting in which we have limited labeled data in the target domain, and will employ parameter-sharing and representation-based methods to induce the transfer. We adapt the setting in [17] to benchmark a broad range of methods while extending it to a novel setting to transfer knowledge between common and rare cancer types.

3 Method

3.1 Classic Baselines

Following prior work, we choose to evaluate the performance of $l1$-logistic regression [32] and XGBoost [33] on our benchmarks. Logistic regression is implemented with the scikit-learn library [34], and MultiOutputClassifier is used in multilabel classification settings. Stacking is implemented for the fine-tuned models [17]. Specifically, the outputs from the source classifiers are augmented to the input of the finetuning model when training on the target domain dataset.

3.2 Deep Learning Baselines

For our deep learning models, we evaluate the performance of one competitive hybrid model called Neural Oblivious Decision Ensembles (NODE) [14], as well as more recent attention-based model architectures such as AutoInt [35], TabTransformer [12], FT Transformer [13], and TabNet [11]. All architectures are implemented with the PyTorch Tabular library [36], and further adapted for transfer learning. To finetune, we transfer the model weights for the backbone and embedding layer from the source model, but reinitialize and train the prediction head.

3.3 Training

For both the source and target domains, we split the data into train-validation-test datasets (Table 1). The source models are all trained on the source train dataset and hyperparameter-tuned on the source validation dataset. The source models are evaluated on the source test split. The "target models" are trained and tuned from scratch on the target train and validation datasets. The "finetune models" inherit the frozen encoder from the source models, and are further tuned on the target train and validation datasets. Both the finetune and target model performances are evaluated on the target test dataset. Hyperparameter tuning was performed with Optuna [37] over 30 trials. We optimized for validation F1 score on the MetaMIMIC dataset and for validation AUC on the TCGA dataset. We set the hyperparameter configurations such that the minimum and maximum number of trainable parameters across all deep models ranges between $1e5$ and $1e7$ parameters.

	metaMIMIC			TCGA		
	train	val	test	train	val	test
source	22,701	5,239	6,985	4,007	567	1,151
target	200	200	6,985	508	72	146

Table 1: The sample sizes after the dataset split for the two transfer learning experiments.

4 Datasets

4.1 MetaMIMIC

MetaMIMIC repository [38] is built upon the MIMIC-IV clinical database [39], a compilation of anonymized patient data originating from the Beth Israel Deaconess Medical Center ICU admissions. We adapt the setting established in [17] and define 12 binary prediction tasks, each corresponding to different medical diagnoses such as hypertensive diseases, ischemic heart disease, diabetes, and alcohol dependence, among others. This dataset encompasses records from 34,925 patients and 172 distinct features. The feature set comprises a range of statistics, including the mean, maximum, and minimum values of laboratory test results, as well as general patient attributes like height, weight, age, and gender. Gender is treated as a categorical variable, while the remaining 171 features are treated as continuous features. We follow the same data preprocessing as [38], for the specific limited data transfer setting of 200 target training labels (Table 1).

4.2 The Cancer Genome Atlas (TCGA)

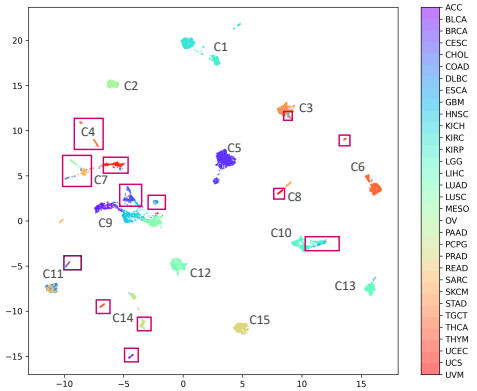


Figure 1: UMAP visualization of the preprocessed mRNA Seq gene expression input, colorized by the 32 cancer types. General groupings (C1 to C15) are labeled manually according to the spatial organization of the manifold. The regions of rare cancer types are boxed in magenta.

The TCGA dataset [40] encompasses genomics data from more than 20,000 primary cancer samples over 33 cancer types that have either poor prognosis or high public health impact. We define the prediction task as a binary classification of mortality from the patients' gene expression profiles. Gene expression profiles are preprocessed into the form of mRNA Seq (log norm). There are 32 cancers with sufficient data for the defined mortality prediction task. The raw gene expression data is very high dimensional. We performed feature selection with XGBoost for the mortality prediction task and selected 2,400 relevant genes out of the available 58,581 genes. The AUC was 0.752 with 58,581 genes and 0.753 with 2,400 relevant genes, demonstrating that the selected genes are well sufficient for the proposed task.

To define the transfer learning setting, we categorized the cancers into rare and common cancers with the aid of Orphanet [41], a comprehensive inventory of rare diseases with their phenotypic features and available treatment. In the "Rare" column in Figure 2, we identify the corresponding Orphanet code for cancers that are categorized as rare by the inventory. The patient cohort is further separated into the train, validation, and test sets with a 70-10-20 split.

Abbr	Cancer Name	Group	Rare
ACC	Adrenocortical carcinoma	C14	ORPHA:1501
BLCA	Bladder urothelial carcinoma	C9	
BRCA	Breast invasive carcinoma	C5	
CESC	Cervical squamous cell carcinoma, endocervical adenocarcinoma	C9	ORPHA:213767
CHOL	Cholangiocarcinoma	C11, C13	ORPHA:70567
COAD	Colon adenocarcinoma	C11	
DLBC	Lymphoid neoplasm diffuse large B-cell lymphoma	C7	ORPHA:544
ESCA	Esophageal carcinoma	C3, C9	ORPHA:70482
GBM	Glioblastoma multiforme	C10	ORPHA:360
HNSC	Head & Neck squamous cell carcinoma	C9	
KICH	Kidney Chromophobe	C1	
KIRC	Kidney renal clear cell carcinoma	C1	
KIRP	Kidney renal papillary cell carcinoma	C1	
LGG	Brain Lower Grade Glioma	C10	
LIHC	Liver hepatocellular carcinoma	C13	

Abbr	Cancer Name	Group	Rare
LUAD	Lung adenocarcinoma	C12	
LUSC	Lung squamous cell carcinoma	C9, C12	
MESO	Mesothelioma	C7	ORPHA:50251
OV	Ovarian serous cystadenocarcinoma	C2	
PAAD	Pancreatic adenocarcinoma	C14	
PCPG	Pheochromocytoma, Paraganglioma	C14	ORPHA:573163
PRAD	Prostate adenocarcinoma	C15	
READ	Rectum adenocarcinoma	C11	
SARC	Sarcoma	C7	ORPHA:223727
SKCM	Skin Cutaneous Melanoma	C8	
STAD	Stomach adenocarcinoma	C3	
TGCT	Testicular Germ Cell Tumors	C4	ORPHA:363504
THCA	Thyroid carcinoma	C6	
THYM	Thymoma	C6, C14	ORPHA:99867
UCEC	Uterine Corpus Endometrial Carcinoma	C7	
UCS	Uterine Carcinosarcoma	C7	ORPHA:213610
UVM	Uveal Melanoma	C8	ORPHA:39044

Figure 2: The 32 cancers with their abbreviations, full name, grouping according to the UMAP in Figure 1, and the Orphanet code if the cancer is rare.

5 Results

5.1 MetaMIMIC Transfer Learning Performance

AUC	source		finetune		target		diff
models	**mean**	**CI**	**mean**	**CI**	**mean**	**CI**	
XGBoost	0.827	(0.824, 0.831)	0.723	(0.709, 0.738)	0.714	(0.699, 0.729)	0.009
LogReg	0.823	(0.820, 0.826)	0.678	(0.664, 0.693)	**0.731**	(0.717, 0.744)	-0.053
AutoInt	**0.833**	(0.829, 0.836)	0.690	(0.673, 0.705)	0.701	(0.686, 0.716)	-0.011
NODE	**0.833**	(0.830, 0.837)	**0.754**	(0.740, 0.769)	0.722	(0.707, 0.736)	0.032
FTTrans	0.832	(0.829, 0.836)	0.712	(0.696, 0.728)	0.689	(0.673, 0.705)	0.023
TabTrans	0.832	(0.829, 0.835)	0.684	(0.669, 0.700)	0.674	(0.658, 0.689)	0.010
TabNet	0.811	(0.808, 0.816)	0.549	(0.531, 0.567)	0.510	(0.493, 0.527)	**0.039**

F1	source		finetune		target		diff
models	**mean**	**CI**	**mean**	**CI**	**mean**	**CI**	
XGBoost	**0.519**	(0.512, 0.528)	0.414	(0.393, 0.436)	0.400	(0.378, 0.423)	0.014
LogReg	0.479	(0.471, 0.486)	0.387	(0.366, 0.409)	0.261	(0.245, 0.280)	**0.126**
AutoInt	0.504	(0.496, 0.511)	0.319	(0.299, 0.338)	0.460	(0.441, 0.480)	-0.141
NODE	0.483	(0.475, 0.491)	**0.476**	(0.455, 0.495)	**0.483**	(0.464, 0.504)	-0.007
FTTrans	0.507	(0.500, 0.515)	0.418	(0.400, 0.437)	0.478	(0.460, 0.496)	-0.060
TabTrans	0.500	(0.492, 0.508)	0.390	(0.369, 0.410)	0.434	(0.416, 0.453)	-0.044
TabNet	0.460	(0.452, 0.468)	0.326	(0.311, 0.342)	0.360	(0.348, 0.373)	-0.034

Table 2: The ROC-AUC (top) and F1 (bottom) scores of all models averaged over the 12 conditions for the MetaMIMIC task. The diff row is the finetune subtracted by the target performance. The 95 confidence intervals (CI) are computed through bootstrapping for 100 iterations.

Based on the AUC benchmarking results (Table 2, top), we see that overall NODE is the best-performing model. TabNet does not perform well, especially in the target domain. All deep models except for TabNet perform better in terms of the average AUC than the classic models in the source domain. For the target domain, the classic models perform very well. Only NODE (AUC=0.722) performs better than the classic models (AUC=0.714 for XGBoost and AUC=0.731 for LogReg). We hypothesize that the deep models suffer more than the classic models from the limited data regime of 200 samples during training. We can observe the effect of transfer learning through the diff row, which shows the difference in performance between the finetune and target models. We note that except for logistic regression and AutoInt, all models have positive transfer. TabNet is the model with the greatest positive transfer, but the AUC value of 0.510 is a very easy baseline to improve upon. Besides TabNet, NODE is the model with the second most positive transfer (diff=0.032).

In terms of the F1 score (Table 2, bottom), XGBoost and NODE have relatively higher F1 scores across source, finetune, and target models. Interestingly, the F1 scores decrease for deep models with transfer, while F1 scores increase for classic models with transfer. Specifically, we find in our analysis that the recall of finetune deep models is lower than target deep models. For instance, AutoInt has an average recall of 0.316 for finetune models and an average recall of 0.598 for target models.

5.2 Analysis of Transfer across Individual MetaMIMIC Conditions

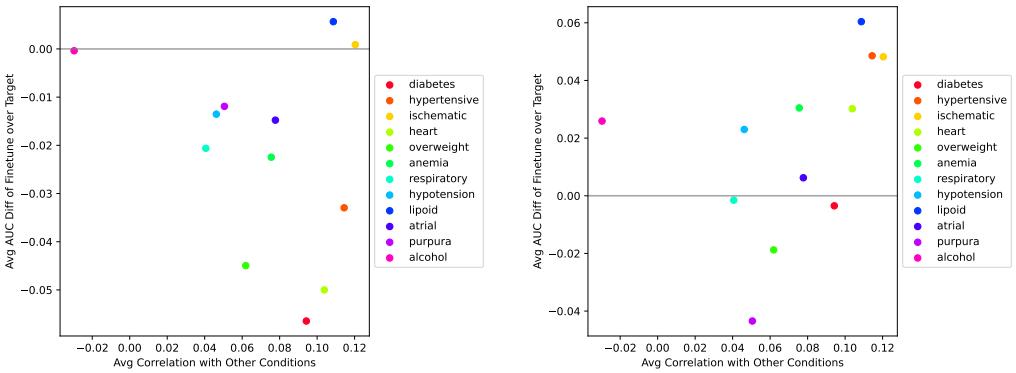


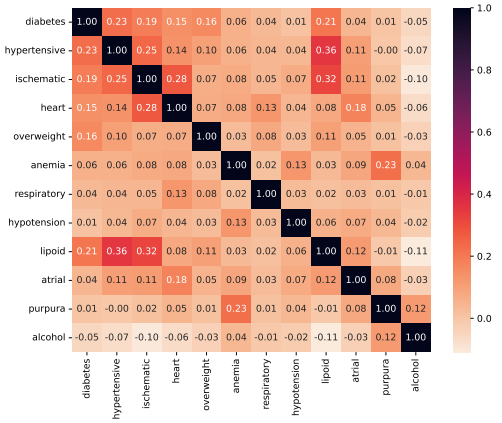
Figure 3: The correlation versus the increase in AUC performance of the finetuned models over the target models for the 12 conditions of MetaMIMIC, split and averaged by classic models (left) and deep models (right).

In Figure 3, we visualize the association between the correlation of a target condition with other source conditions and the success of the transfer. The x-axis is the average Pearson correlation of each particular target condition with the remaining 11 conditions. The y-axis is the average AUC performance of the finetune models subtracted by the target models. The left plot takes the AUC average over classic models, while the right plot takes the AUC average over the deep models. If we analyze the association in the scatterplots, we get a correlation of -0.253 for the classic models and a correlation of 0.395 for the deep models. For deep models, we observe that target conditions such as "Disorders of lipoid mechanism" (lipoid) and "Ischematic heart disease" (ischematic) with higher correlation with other source conditions tend to transfer more easily, as expected. As shown in Figure 4a, lipoid has 0.36 correlation with hypertensive and 0.32 correlation with ischematic. Indeed, lipoid has positive transfer, even for classic models. A condition that doesn't follow this trend is "Alcohol dependence" (alcohol). The correlation with other conditions is low, but models obtain the same or better performance in the transfer. We hypothesize that this may be due to the task being inherently easier to predict.

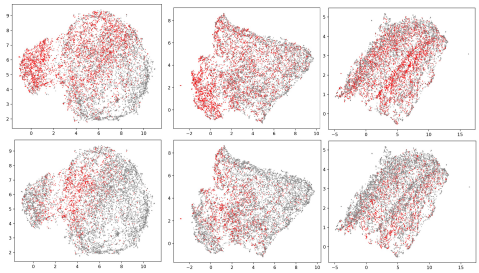
5.3 MetaMIMIC Analysis of Feature Embeddings in Deep Models

In Figure 4b, we visualize how the latent representation of a particular deep model changes across source, finetune, and target training. We choose to analyze the NODE model, as it generally performs well for transfer learning. We analyze the particular set of NODE models associated with the target domain task of predicting "Disorders of lipoid metabolism", since this condition has significant correlations with other source conditions. We visualize the latent representation of the embedding output of the NODE backbone, which is the last layer of the model subcomponent that is transferred in our transfer learning setup.

We see in the top row of Figure 4b that the NODE model is able to distinguish between lipoid and non-lipoid patients for the source, finetune, and target models. In the source model (top left figure), the lipoid patients are concentrated in the left region and upper left region of the right region. We see that this latent representation is morphed during the fine-tuning process (top middle figure), but the concentrated regions of lipoid patients remain largely the same. For the model trained from scratch on the target dataset (top right figure), the patients are instead concentrated on the "edges" of the two stacked regions.



(a) Heatmap of Pearson correlation matrix of the 12 conditions in the MetaMIMIC task.



(b) UMAPs of the embedding layers of the set of NODE models for transfer to downstream target label of "Disorders of lipoid metabolism" (lipoid).
Right to left: Source finetune, target models.
Top row: lipoid patients in red.
Bottom row: heart failure patients in red.

On the bottom row, we see that the NODE model distinguishes between heart failure patients and non-heart failure patients for the source and finetune models, but less so for the target model. In the source and finetune representations, the heart failure patients are concentrated around the central region connecting the left and right regions of the UMAP. In contrast, since the target model is not trained to predict heart failure, there is no noticeable pattern in the learned embedding space, and heart failure patients are dispersed throughout the manifold. From these visualizations, we conclude that the finetune model is able to preserve learned patterns from the source model, which induces its representation space to be more expressive than the target models trained from scratch on a single condition.

5.4 MetaMIMIC Analysis of Important Features in Classic Models

diabetes	**hypertensive**	**ischematic**	**heart**
max_Glucose	**Pred_lipoid**	**Pred_lipoid**	avg_GCS (Eye)
avg_Glucose	**Pred_diabetes**	**Pred_diabetes**	avg_systolic BP
avg_Creatinine	**Pred_heart**	max_Braden Activity	**Pred_diabetes**
avg_GCS (Eye)	**Pred_ischematic**	**Pred_heart**	avg_Braden Mobility
avg_GCS (Verbal)	avg_systolic BP	max_Braden Friction	avg_GCS (Verbal)

overweight	**anemia**	**respiratory**	**hypotension**
max_O2 Flow	avg_Hemoglobin	avg_pCO2	min_systolic BP
avg_pCO2	min_Hemoglobin	avg_Bicarbonate	avg_BUN
max_pH	min_Hematocrit	min_Glucose	avg_Potassium
Pred_diabetes	avg_Hematocrit	age	avg_MCHC
first_Admission Weight (Kg)	avg_Base Excess	avg_O2 sat	min_Pain Level

lipoid	**atrial**	**purpura**	**alcohol**
max_GCS (Verbal)	avg_GCS (Motor)	avg_Platelet Count	min_Platelet Count
max_pH	avg_systolic BP	min_Platelet Count	min_Red Blood Cells
Pred_ischematic	max_MCHC	avg_Heart Rate	avg_Platelet Count
max_SpO2 Desat Limit	min_Calcium, Total	min_Platelet Count	max_Creatinine
min_Urea Nitrogen	min_AST	max_GCS (Motor)	avg_Magnesium

Table 3: The top five most important features of the XGBoost model with stacking for fine-tuning. Each condition column is the target domain to transfer to. The stacked features from the output of the source model are bolded. Preprocessed laboratory values are prefixed with the average (avg), minimum (min), maximum (max), or first entry (first).

From the feature importance analysis in Table 3, we note that for certain conditions, the XGBoost models exploit the class prediction logits of the stacked source model for prediction on the target task. "Hypertensive diseases" (hypertensive) and "ischematic heart disease" (ischematic) both heavily utilize the source model logits for the correlated conditions of lipoid, diabetes, and heart failure as important predictive features. Similarly, heart failure utilizes the prediction for diabetes and lipoid utilizes the prediction for ischematic. Although diabetes is reasonably well correlated with this set of conditions as well (Figure 4a), the model prefers to use more direct features such as glucose and creatinine levels (reflective of kidney damage induced by diabetic complications). Thus, although the transfer of knowledge is not as nuanced as latent patterns for the case of deep models, the classical models can benefit from transfer in the form of stacking, especially for correlated tasks.

5.5 TCGA Transfer Learning Performance

AUC	**source**		**finetune**		**target**		
models	**mean**	**CI**	**mean**	**CI**	**mean**	**CI**	**diff**
XGBoost	**0.742**	(0.713, 0.774)	0.787	(0.720, 0.865)	**0.788**	(0.722, 0.864)	-0.001
LogReg	0.696	(0.663, 0.727)	0.756	(0.682, 0.829)	0.756	(0.683, 0.828)	0.000
AutoInt	0.729	(0.702, 0.758)	0.785	(0.710, 0.857)	0.774	(0.694, 0.844)	0.011
NODE	0.736	(0.710, 0.763)	**0.804**	(0.736, 0.869)	0.772	(0.705, 0.837)	**0.032**
FTTrans	0.725	(0.696, 0.752)	0.670	(0.578, 0.774)	0.762	(0.689, 0.826)	-0.092
TabTrans	0.735	(0.702, 0.766)	0.783	(0.707, 0.855)	0.785	(0.716, 0.860)	-0.002
TabNet	0.649	(0.609, 0.673)	0.636	(0.545, 0.712)	0.621	(0.524, 0.702)	0.015

F1	**source**		**finetune**		**target**		**diff**
models	**mean**	**CI**	**mean**	**CI**	**mean**	**CI**	
XGBoost	**0.823**	(0.799, 0.847)	**0.837**	(0.792, 0.890)	0.817	(0.763, 0.871)	0.020
LogReg	0.797	(0.775, 0.819)	0.797	(0.739, 0.850)	0.797	(0.739, 0.850)	0.000
AutoInt	0.751	(0.728, 0.776)	0.792	(0.742, 0.841)	0.816	(0.742, 0.878)	-0.024
NODE	0.753	(0.727, 0.779)	0.785	(0.714, 0.833)	0.775	(0.706, 0.837)	0.010
FTTrans	0.782	(0.758, 0.806)	0.820	(0.767, 0.868)	0.731	(0.652, 0.799)	**0.089**
TabTrans	0.767	(0.745, 0.789)	0.807	(0.752, 0.868)	**0.824**	(0.771, 0.881)	-0.017
TabNet	0.699	(0.674, 0.726)	0.676	(0.605, 0.744)	0.688	(0.607, 0.764)	-0.012

Table 4: The ROC-AUC (top) and F1 (bottom) scores of all models for the TCGA mortality prediction task. The diff row is the finetune subtracted by the target performance. The 95 confidence intervals (CI) are computed through bootstrapping for 100 iterations.

In Table 4, we observe that XGBoost, NODE, and TabTransformer all perform reasonably well for the mortality prediction task on gene expression data from TCGA. TabNet's overall performance on both source and target domains is subpar, with AUCs all below 0.7. NODE has the greatest gain in AUC performance due to transfer learning (diff=0.032). The classic models do not benefit from stacking for transfer learning. FT-Transformer's performance decreases significantly due to the transfer (diff= -0.092). Interestingly, most models perform better on mortality prediction in the target domain than in the source domain. There is a 29% mortality rate for the common cancers and a 33% mortality rate for the rare cancers. We hypothesize that the mortality prediction may be inherently easier on the rarer cancers, although further analysis with clinical input is needed.

5.6 TCGA Analysis of Feature Embeddings in Deep Models

We visualize the latent representations of the last transferred layer of the deep NODE model across the source, finetune, and target settings in Figure 5. We do not observe any significant global patterns of mortality across the clusters. Each subtype of cancer has its own subgroup of patients who died. In general, the unseen dataset gets mapped to the same region as the dataset the model was trained on. The NODE embeddings have less distinct clusters than the clustered TCGA gene expression input. The finetune model embeddings look more similar to the target embeddings than the original source embeddings, as the clustering patterns of rare cancers differ significantly from the common cancers. Despite the lack of overarching global patterns, however, the model is able to perform very

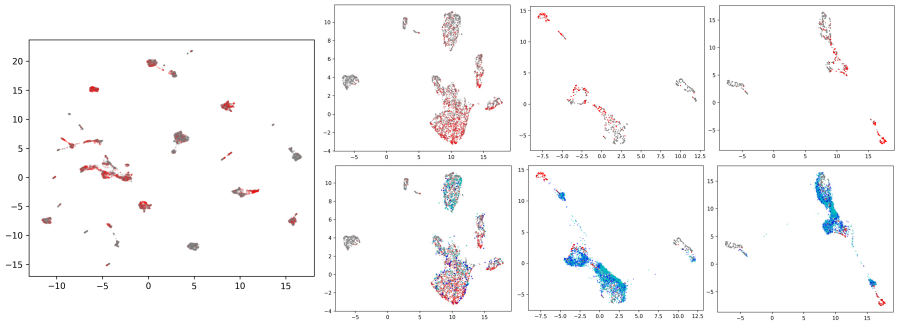


Figure 5: UMAPs of the embedding layers of the set of NODE models for TCGA mortality prediction, transfer from common to rare cancers.
Left: UMAP of the input TCGA gene expression data. Patients who die are in red and patients who live are in gray.
Right: From left to right: source, finetune, target models. Top row are UMAPs of the model on the dataset it was trained on (dead patients in red, alive in gray). Bottom row are of the same UMAPs as the top row, but overlaid with the unseen dataset (dead in dark blue, alive in cyan).

well on the task of mortality prediction (Table 4). Perhaps the distinction is more noticeable in the downstream prediction head, or local patterns not easily recognizable in a UMAP are responsible for the model performance.

5.7 TCGA Analysis of Important Features in Classic Models

Source	Finetune	Target
SLC2A1	**NASP**	AUNIP
INPP5J	PAPSS2	EFNA3
IGF2BP3	HAS1	HOXA11
RRAD	RP3-426I6.2	INHBA
FAM72B	RP11-803D5.4	**NASP**

Table 5: The top five most important genes for the XGBoost model on the TCGA mortality task prediction.

Given the mixture of cancer types within source and target domains and the nuanced relationships between the expression of correlated genes, the top important genes for an XGBoost model may differ significantly from run to run. In Table 5, we see that for the top five important genes, there is only one overlap of the NASP gene, which is present in both the finetune and target models. NASP is known in the literature to have higher expression levels for liver cancer [42] and ovarian cancer [43]. The other top important genes also typically are expressed in higher levels in specific cancer tissue. Lastly, we note that the finetune model does not seem to make use of the stacked mortality predictions from the source model. The prediction logits do not appear within the top 100 important features for multiple runs of the model.

6 Conclusion

In this paper, we have benchmarked the classic methods such as XGBoost and logistic regression and deep learning methods such as AutoInt, NODE, FT-Transformer, TabTransformer, and TabNet for two transfer learning datasets within the domain of tabular data. We expand upon the benchmarking results of a previously proposed transfer setting of MetaMIMIC with new models and provide additional detailed insights into the mechanisms of transfer learning of deep and classic models for each disease condition. We further propose and benchmark a new transfer learning setting that utilizes high-dimensional genetic data with the TCGA dataset.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] Xuhong Li, Yves Grandvalet, Franck Davoine, Jingchun Cheng, Yin Cui, Hang Zhang, Serge Belongie, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Transfer learning in computer vision tasks: Remember where you come from. *Image and Vision Computing*, 93:103853, 2020.

[3] Andrzej Brodzicki, Michal Piekarski, Dariusz Kucharski, Joanna Jaworek-Korjakowska, and Marek Gorgon. Transfer learning methods as a new approach in computer vision tasks with small datasets. *Foundations of Computing and Decision Sciences*, 45(3):179–193, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18, 2019.

[6] Fuchao Yu, Xianchao Xiu, and Yunhui Li. A survey on deep transfer learning and beyond. *Mathematics*, 10(19):3619, 2022.

[7] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[8] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022.

[9] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

[10] J Yoon. Vime: Extending the success of self- and semi-supervised learning to tabular domain. `https://github.com/jsyoon0823/VIME`, 2020.

[11] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[12] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[13] 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.

[14] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

[15] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[16] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR, 2023.

[17] Roman Levin, Valeriia Cherepanova, Avi Schwarzschild, Arpit Bansal, C Bayan Bruss, Tom Goldstein, Andrew Gordon Wilson, and Micah Goldblum. Transfer learning with deep tabular models. *arXiv preprint arXiv:2206.15306*, 2022.

[18] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1): 43–76, 2020.

[19] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[20] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Ganesh Ramakrishnan, Micah Goldblum, Colin White, et al. When do neural nets outperform boosted trees on tabular data? *arXiv preprint arXiv:2305.02997*, 2023.

[21] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[24] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[25] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[26] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[27] Jannik Kossen, Neil Band, Clare Lyle, Aidan N Gomez, Thomas Rainforth, and Yarin Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34:28742–28756, 2021.

[28] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[29] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[30] Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. Xtab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.

[31] Soma Onishi, Kenta Oono, and Kohei Hayashi. Tabret: Pre-training transformer-based tabular models for unseen columns. *arXiv preprint arXiv:2303.15747*, 2023.

[32] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[33] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[34] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[35] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1161–1170, 2019.

[36] Manu Joseph. Pytorch tabular: A framework for deep learning with tabular data, 2021.

[37] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[38] Mateusz Grzyb, Zuzanna Trafas, Katarzyna Woźnica, and Przemysław Biecek. metamimic: analysis of hyperparameter transferability for tabular data using mimic-iv database. `https://github.com/ModelOriented/metaMIMIC/blob/main/preprint.pdf`, 2021.

[39] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020.

[40] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

[41] Steffanie S Weinreich, R Mangon, JJ Sikkens, ME En Teeuw, and MC Cornel. Orphanet: a european database for rare diseases. *Nederlands tijdschrift voor geneeskunde*, 152(9):518–519, 2008.

[42] Xuan Kang, Yun Feng, Zhixue Gan, Shiyang Zeng, Xiaobo Guo, Xirui Chen, Ye Zhang, Chen Wang, Kuinan Liu, Xuelin Chen, et al. Nasp antagonize chromatin accessibility through maintaining histone h3k9me1 in hepatocellular carcinoma. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1864(10):3438–3448, 2018.

[43] Rouba Ali-Fehmi, Madhumita Chatterjee, Alexei Ionan, Nancy K Levin, Haitham Arabi, Sudeshna Bandyopadhyay, Jay P Shah, Christopher S Bryant, Stephen M Hewitt, Michael G O'Rand, et al. Analysis of the expression of human tumor antigens in ovarian cancer tissues. *Cancer Biomarkers*, 6(1):33–48, 2010.