A Experimental setup

To aid in reproducing the results, we present technical details regarding our experiments.

Initially, we divided the dataset into training and testing parts, allocating three-quarters of the data for training and the remaining quarter for testing. Subsequently, both the training and testing parts were preprocessed based on the characteristics observed in the training set, ensuring that the models were trained on data representative of the real-world scenarios they would encounter.

To further refine the training process, the training dataset was then split again, this time into training and validation datasets with proportions of four fifths and one fifth, respectively. This resulted in final proportions of the train, valid, and test sets being 12/20, 3/20, and 5/20 of the entire dataset. This split was instrumental in tuning the models and preventing overfitting.

Upon completion of hyperparameter optimization, the training and validation datasets were merged into a single *full_train* dataset. The models then underwent final training on this *full_train* dataset, utilizing the hyperparameters identified as optimal in the previous step. This comprehensive training regime, culminating in testing on the separate test split, was designed to mitigate any risk of cross-contamination in the results, ensuring the integrity and reliability of our findings.

Hyperparameter optimization was performed using the PyHopper library, executing 50 optimization steps with four running in parallel and a seeding ratio of 0.5. This optimization was carried out on the train/validation splits, allowing us to fine-tune the models for optimal performance. We used the following ranges of hyperparameters for each method:

LightGBM

```
num_leaves = choice(2, 4, 8, 16, 32, 64),
max_depth = choice(-1, 2, 4, 8, 16, 32, 64),
learning_rate =float(0.001, 0.1, log=True),
n_estimators = choice(10, 50, 100, 200, 500,
1000)
```

XGBoost

```
n_estimators = int(50, 1000, multiple_of=50,
init=50),
max_depth = choice(2, 3, 5, 10, 15),
learning_rate = float(1e-5,1e-1, log=True),
min_child_weight = choice(1, 2, 4, 8, 16, 32),
gamma = choice(0, 0.001, 0.1, 1)
```

Random Forest

```
n_estimators = int(50, 3000, multiple_of=50),
max_features = choice(None, 'sqrt', 0.2, 0.3,
0.5, 0.7),
criterion = choice('gini', 'entropy'),
max_depth = choice(None, 2, 4, 8, 16)
```

Gradient Boosting

```
n_estimators = int(50, 3000, multiple_of=50,
init=50),
max_depth = choice(2, 3, 5, 10, 15),
learning_rate = float(1e-5,1e-1, log=True)
```

NODE

```
layer_dim = int(64, 1024, power_of=2),
num_layers = int(1, 5),
depth = int(2, 7)
```

VisTabNet

```
LR = float(1e-5, 1e-3, "0.1g"),
PROJ_LR = float(1e-5, 1e-3, "0.1g"),
EPOCHS = int(10, 100, multiple_of=10),
PROJECTIONS = choice(8, 16, 32, 64, 128),
PROJ_DEPTH = choice(1, 2, 3, 4)
```

ResNet

```
EPOCHS = choice(10, 30, 50, 100, 150),
PATIENCE = choice(2, 5, 10, 16, 24, 37)
```

FT

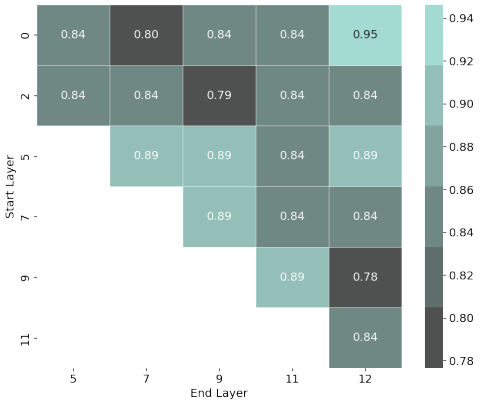
```
N_BLOCK: choice(1,2,3,4,5,6),
D_BLOCK: choice(96, 128, 192, 256, 320, 384),
ATTENTION_DROPOUT: choice(0.1, 0.15, 0.2,
0.25, 0.3, 0.35),
FFN_DROPOUT: choice(0.0, 0.05, 0.1, 0.15, 0.2,
0.25),
EPOCHS = choice(50, 100, 150),
PATIENCE = choice(2, 10, 16, 37)
```

The experiments were conducted on 20 datasets, which are summarized in Table 1.

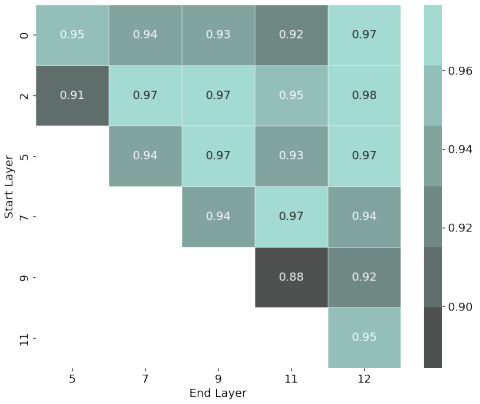
B Detailed results

Figure 1 presents the MCC score of VisTabNet when only the part of the ViT encoder was transferred to VisTabNet architecture. Other layers were completely removed.

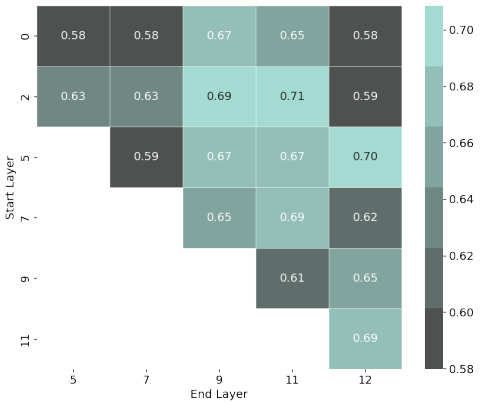
Figure 2 presents MCC scores across training epochs for 5 datasets. Red color indicates the phase of training adaptation and classification networks while blue color shows the fine-tuning phase of the entire model (including Vit encoder). As can be seen the learning rate has to be carefully scheduled to avoid drops in performance.



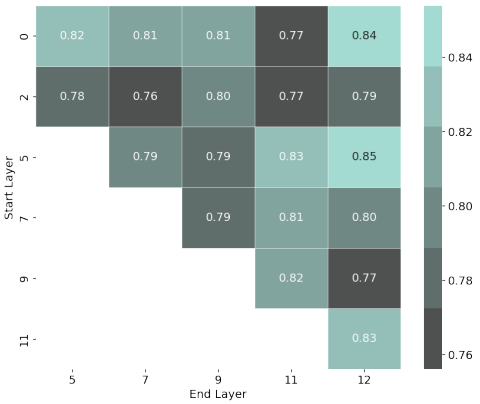
(a) ZOO



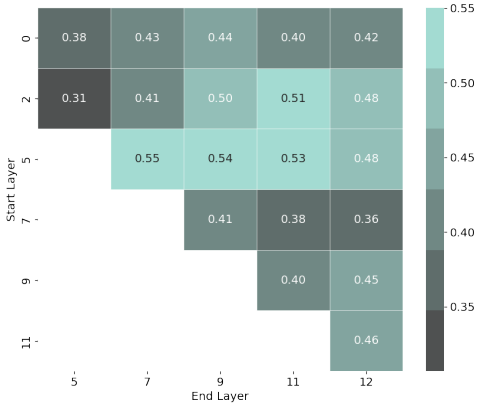
(b) Dermatology



(c) Credit Approval

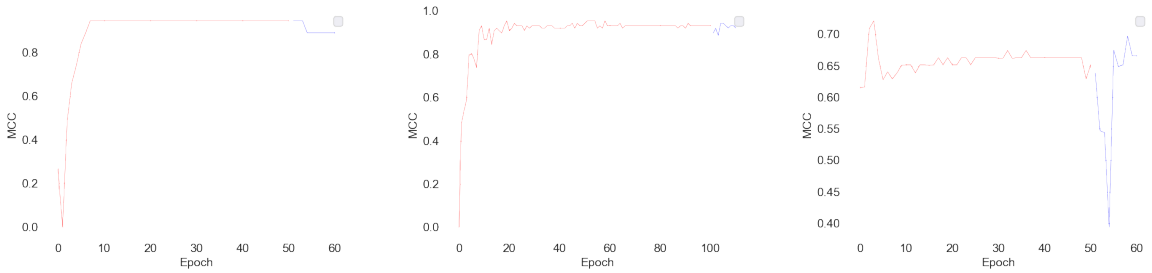


(d) Libras



(e) Cylinder Bands

Figure 1: Performance of VisTabNet when selected layers were removed from the ViT encoder.



(a) ZOO (b) Dermatology (c) Credit Approval

(d) Libras (e) Cylinder Bands

Figure 2: Learning curves across training epochs of VisTabNet. Red color indicates the phase of training adaptation and classification networks while blue color shows the fine-tuning phase of the entire model (including Vit encoder).

Table 1: Summary of the datasets.

Dataset	Size	Continuous Attributes	Categorical Attributes	Classes
Blood Trans.	748	4	1	2
BC Wisconsin	569	30	0	2
Breast Cancer	286	0	9	2
Connectionist	208	60	0	2
Congr. Voting	435	0	16	2
Credit Approval	690	6	9	2
Cylinder Bands	512	20	19	2
Dermatology	366	34	0	6
Ecoli	336	5	0	8
Glass	214	9	0	6
Haberman	306	3	0	2
Horse Colic	368	8	19	2
Ionosphere	351	34	0	2
Libras	360	90	0	15
Lymphography	148	18	0	4
Mammographic	961	1	5	2
Primary Tumor	330	0	17	21
Sonar	208	60	0	2
Statlog Australian	690	5	9	2
Statlog German	1000	23	0	2
Statlog Heart	270	6	7	2
Vertebral	310	6	0	2
Zoo	101	16	0	7