School of Informatics

Informatics Research Review
Deep Learning (DL) vs Gradient Boosted Decision Trees (GBDT) for Tabular Data

B196696
January 2022

Abstract

DL has proved its mettle on image, text and audio datasets. Yet for tabular datasets, the question remains who's the real winner? DL or GBDT - a reigning champion of Kaggle competitions with tabular datasets. It has been answered in 3 stages. First, Multilayer Perceptron (MLP - basic DL strategy for tabular datasets) is compared with GBDT upon several datasets in which GBDT was observed as the winner mostly; second, reasons for the ineffectiveness of DL on tabular datasets were investigated; finally, recently developed DL architectures are analysed which have claimed to outperform GBDT but again GBDT was observed as the winner in most cases.

Date: Thursday 27th January, 2022

Supervisor: Pavlos Andreadis

1 Introduction

DL methods have been very successful for homogeneous datasets like image [He et al., 2016], text [Lai et al., 2015] and audio [Amodei et al., 2015]. Due to their huge success for these datasets [Goodfellow et al., 2016], researchers have moved towards developing more and more advanced techniques for such datasets. Apart from these, there exists an another more common type of datasets as well, tabular datasets. We can define a tabular dataset as a table which contains n numerical columns $N_1, ..., N_n$ and c categorical columns $C_1, ..., C_c$. All columns are random variables following a joint distribution $P(N_{1:N}, C1 : c)$. Each data point can be understood as a row in the table, or – taking a probabilistic view – as a sample from the unknown joint distribution [Borisov et al., 2021]. These tabular datasets are heterogeneous in nature which suggests that each columns values are coming from unrelated sources, thus they are of different scales, different datatypes such as numeric or categorical (binary, nominal, ordinal, high cardinal ordinal). Most of the real world datasets are of this formats only, such as Life sciences [Czerniak and Zarzycki, 2003], Physical sciences [Evans and Fisher, 1994], CS/Engineering [Stolfo et al., 1999], Game [Cattral et al., 2002] and so forth. Even though most of the data that we have in real world is in tabular form [Chui,], researchers have explored very less for DL in the field of tabular dataset and GBDT methods continued being the reigning champion for tabular datasets [Kaggle,].

But then the question arises that why we are even worried about using DL models for Tabular datasets? One of the obvious advantages of using Deep Learning is the expected performance increase with an increase in data size [Hestness et al., 2017]. Deep Learning also facilitates an end to end gradient-based learning with several benefits:(i)bridge the gap between homogeneous and heterogeneous data i.e, training text data and tabular data together for ex. chat-bot which takes text as input but also take demographic information of person as input in form of tabular data to give customised experience; (ii) removes the need for feature engineering; (iii) learning from continuous streaming data; (iv) transfer learning for domain adaptation [Goodfellow et al., 2016]; semi supervised learning [Wang et al., 2021] and generative modeling [Radford et al., 2016]. Apart from these, DL methods are also valuable for AutoML. They can be used for multimodal learning problems [Baltrušaitis et al., 2019], for tabular data distillation [Medvedev and D'yakonov, 2020] and for federated learning [Roschewitz et al., 2021].

Recently there have been several attempts in DL domain to outperform GBDT for tabular datasets. But due to the absence of standard benchmark datasets like ImageNet [Deng et al., 2009] for images or GLUE [Wang et al., 2018] for texts, It's quite hard to compare the DL methods (which have several advantages and are the state of the art for homogeneous data) with GBDT (the current state of the art for tabular datasets) for the tabular datasets (most abundant form of data). This entire work deals with the same one major question: who is better? DL or GBDT for tabular datasets. First, this work compared the basic DL architecture - MLP with GBDT for several tabular datasets and it has been found that in most of the cases GBDT continues to be the winner. Then, a second stage of comparison is done for all of the recent architectures or pipelines in DL domain which have promised that they have outperformed GBDT. This second stage of comparison is needed as each of these new approaches have used different datasets to mark their superior performance. So that's why, few of the most successful (at-least on their original paper) and recent architectures have been compared here with GBDT on the datasets which were not used in the original paper to verify their superiority over GBDT as claimed in the original paper. In addition, a detailed analysis for possible reasons of ineffectiveness of DL over tabular datasets also has been carried out. The main motive of this survey is to guide anyone who wants to propose a new architecture in DL for a tabular dataset by giving a baseline

in terms of comparison of MLP vs GBDT, possible reasons for ineffectiveness of DL on tabular datasets so that one can work to solve one of these issues and also giving one a right track to think of possible ways of introducing new architecture/pipeline in DL domain.

This entire survey is organised as follows: To introduce the reader to basic DL architecture (MLP) and GBDT, a quick description of their working mechanisms have been provided in section 2. Section 3 compares the basic DL architecture (MLP) specifically designed for tabular datasets with GBDT and also lay down the possible reasons for ineffectiveness of DL over tabular datasets. Section 4 mentions the recent efforts in DL domain to outperform GBDT. Then, a second stage comparison of these new architectures as discussed above has been done in section 5 followed by a summary and conclusion in section 6 and 7 respectively.

2 Basic mechanism of DL and GBDT

2.1 MLP working principle

MLP is the most basic DL architecture specially designed for tabular datasets. It's made up of 3 types of layers - the input layer, the output layer and a set of hidden layers between the two. The inputs are fed from the input layer, hidden layer are responsible for feature engineering and the output layer is responsible for making predictions. Each layers consists of set of neurons connected with each other through weights. These weights are driving factor of the MLP and are adjusted according to the given data using back-propagation and gradient descent.

2.2 GBDT working principle

GBDT and Random Forest are both Ensemble learning techniques that perform regression or classification tasks by combining the outputs of individual decision trees. But both of these methods differs in the way individual trees are built and the way the results are combined. Random forest built decision trees independent from each other and combined them parallely by averaging or voting. On the other hand, GBDT uses a method called boosting which refers to combining each weak learners sequentially so that each new tree corrects the error of previous ones. Weak Learners are usually decision trees with only one split called decision stumps. GBDT sequentially combines weak learners to fit to the residuals from previous step so that the overall model improves. The final model aggregates the results along the way from each step and finally a strong learner is achieved. A loss function (MSE for regression or log loss for classification) is used to detect the residuals. Another important point to note that When a new tree is added it doesn't change the previous tree.

3 Stage 1 Comparison: DL (MLP) vs GBDT

3.1 Comparison across various datasets

At the stage 1 comparison, main motive is to decide whether base DL architecture for tabular datasets (MLP) is continuously outperformed by GBDT or not?

Experiment 1: Something similar was observed in [Gorishniy et al., 2021] where author build the prediction model for regression and classification tasks on the datasets described in Table 1. Same train-test-validation stratified split and same hyper-parameter tuning protocol has

Name	Abbr	# Train	#Validation	# Test	# Num	# Cat	Task type	Batch size
California Housing	CA	13209	3303	4128	8	0	Regression	256
Adult	AD	26048	6513	16281	6	8	Binclass	256
Helena	HE	41724	10432	13040	27	0	Multiclass	512
Jannis	JA	53588	13398	16747	54	0	Multiclass	512
Higgs Small	HI	62752	15688	19610	28	0	Binclass	512
ALOI	AL	69120	17280	21600	128	0	Multiclass	512
Epsilon	EP	320000	80000	100000	2000	0	Binclass	1024
Year	YE	370972	92743	51630	90	0	Regression	1024
Covtype	CO	371847	92962	116203	54	0	Multiclass	1024
Yahoo	YA	473134	71083	165660	699	0	Regression	1024
Microsoft	MI	723412	235259	241521	136	0	Regression	1024

Table 1: Datasets description for stage 1 comparison. [Gorishniy et al., 2021]

	CA	**AD**	**HE**	**JA**	**HI**
MLP	0.499±2.9e-3	0.852±1.9e-3	**0.383±2.6e-3**	0.719±1.3e-3	0.723±1.8e-3
XGBoost	**0.433±1.6e-3**	**0.872±4.6e-4**	0.375±1.2e-3	**0.721±1.0e-3**	**0.727±1.0e-3**

	EP	**YE**	**CO**	**YA**	**MI**
MLP	**0.8977±4.1e-4**	**8.853±3.1e-2**	0.962±1.1e-3	0.757±3.5e-4	0.747±3.3e-4
XGBoost	0.8837±1.2e-3	8.947±8.5e-3	**0.969±5.1e-4**	**0.736±2.1e-4**	**0.742±1.3e-4**

Table 2: Test results on tabular datasets for stage 1 comparison. Bold marked represents the best model i.e, lowest RMSE for regression and highest accuracy for classification. [Gorishniy et al., 2021]

been maintained across all datasets. RMSE has been used for regression tasks and accuracy for classification tasks. For each of these metrics calculations, 15 experiments were trained with random seeds and then average test performance was reported. From the results (Table 2), it can be observed that 7 out of 10 times XGBoost (a GBDT model outperformed the MLP for tabular datasets.

Experiment 2: Similar comparison has been done in a recent paper [Arik and Pfister, 2019] across 4 more common datasets, where differences in performances are even more significant (Table 3). Though exact experimentation setup has not been provided in the paper. Out of the 4 datasets,3 times XGBoost has outperformed DL architectures for tabular datasets.

3.2 Ineffectiveness of DL against GBDT

As observed in above section, GBDT are outperforming DL methods for most of the tabular datasets. But as we know DL methods are giving state of the art performance in all other types of datasets image, text and time series. But it's not clear why it's lagging behind in

	Poker hand induction	**Higgs Boson**	**Sarcos**	**Rossmann store sales**
MLP	0.5	**0.7844**	2.13	512.62
GBDT	**0.666**	0.7698	**1.44**	**490.83**
	Classification		**Regression**	

Table 3: Test results on tabular datasets for stage 1 comparison as observed from [Arik and Pfister, 2019]. Bold marked represents the best model i.e, lower MSE for regression and higher accuracy for classification.

Tabular datasets? Possible reasons for ineffectiveness of DL methods for tabular datasets (these reasons are mentioned considering basic DL methods for tabular datasets and not few recent architectures that tried to overcome these reasons):

1. **Inappropriate training data:** Poor data quality is a major issue of a real world tabular datasets. They usually have lots of missing values (for ex. not everyone will give recommendations for every movies in a movie rating datasets) [Sánchez-Morales et al., 2019], extreme outliers [Pang et al., 2021], relatively lower number of data points given high-dimensional feature vectors [Xu and Veeramachaneni, 2018], erroneous or inconsistent data [Karr et al., 2006]. Tabular datasets also usually have a large number of categorical features of different types (binary, nominal, ordinal or high cardinals ordinals) for ex zip code of the city etc. Sometimes the order of these categorical variables are not important i.e, ordinal and also have high cardinality. Careful handling of such variables are needed by learning the embeddings using supervised rate, weight of evidence or Perlich ratio and many others. Such inappropriate training data are in abundance in case of tabular datasets and they can't be trained directly by a DL method, they need proper pre-processing first, unlike homogeneous data like images which can be directly utilised by Convolutional Neural Networks.

2. **Missing required inductive bias:** The mechanisms used by learning algorithms to put some restrictions over the underlying model space to prioritize certain solutions with specific properties are called Inductive biases or Learning biases. Usually there are no spatial correlations between the columns in tabular datasets [Zhu et al., 2021b] or even if there are are some dependencies between features, they are really complex and irregular. Thus, inductive biases, favoring the DL methods over homogeneous datasets like spatial inductive bias for Convolutional Neural Networks or Structured Perception And Relational Reasoning for Deep Reinforcement Learning are not valid for such tabular datasets [Katzir et al., 2021, Rahaman et al., 2019, Mitchell, 2017].

3. **Correlated features:** Correlations between several columns of a tabular dataset is quite common. Usually there exists a smaller set of features that really contribute to the predictive power of the model. Thus inductive bias in this case is that there are highly correlated features, so the best strategy is to select the minimum number of features.

4. **Model Sensitivity:** Tabular datasets are very sensitive to small perturbations of the data [Szegedy et al., 2014, Levy et al., 2020]. The smallest change of a technique for using categorical feature (one hot encoded or a specific embeddings) might have a large effect on the predictions. This is usually not the case with homogeneous (in-fact any continuous) data sets. Unlike MLP, GBDT can handle tiny perturbations very well by selecting a threshold value for a feature and ignoring the rest of the datasets. MLP have high curvature decision boundaries [Poole et al., 2016, Achille et al., 2019] due to this high sensitivity, but GBDT can learn hyper-plane like boundaries. Strong regularization to learn-able parameters [Shavitt and Segal, 2018, Kadra et al., 2021] in MLP is one of the suggested approach to reduce this high input data dependent sensitivity.

4 Recent Efforts in DL to outperform GBDT

There has been several attempts recently in DL domain to outperform GBDT. So, Instead of explaining all recent attempts, all possible categories of such attempts will be studied with few examples of each. All of these efforts can be categorised into following three main categories:

4.1 Pre-processing or specific Encoding

This category include techniques which can transform the tabular or heterogeneous datasets to homogeneous datasets so that the transformed data will also have inductive biases that DL algorithms are able to utilise very well in image based data for example.

One of the approach to do this transformation to tabular dataset is by using a pretty recent approach VIME (Value Imputation and Mask Mutation) [Yoon et al., 2020]. It utilises a novel approach of using self and semi-supervised learning for tabular datasets. It involves training of an encoder which can covert the categorical and numerical features to a homogeneous representation but still maintaining most of the information contained in the original tabular dataset. This transformed output is used as a new input to DL predictive model.

Several attempts have also been made recently to come up with a methodology to convert these tabular datasets to an image data with a motivation of huge success now-a-days of DL methods in computer vision domain. First successful attempt has been made by SuperTML architecture [Sun et al., 2019] which is a simple data conversion technique from tabular data to 2d matrices which can be later then modeled by CNN or several other computer vision DL architectures. Yet another similar paper following this idea introduces a new architecture IGTD (Image generator for tabular data) [Zhu et al., 2021c]. As already discussed a tabular dataset doesn't have spatial relationships between features. In order to add such relationships, this method assigns a feature to pixel values so that similar features are placed close to each other in the image. The optimal assignment is determined by minimizing the difference between the ranking of features distances and the ranking of assigned pixels distances. This idea will be very effective if there exists strong relationship between features but it may not be effective if features are completely independent of each other. Author have used gene expression profiles which have correlated features and thus IGTD may introduce favourable inductive bias for DL methods. One more similar technique which is quite famous is DeepInsight [Zhu et al., 2021a] which works bit differently than these two. It first applies PCA or tSNE or any such dimensionality reduction techniques and then determine top two features. Plot of these top 2 features designed by convex hull algorithm are then used as an image by a DL method from computer vision domain.

4.2 Architectural Improvement

Most of the promising developments in the DL for tabular dataset are done in this category. This category focuses on developing an entire new DL method specifically for tabular or heterogeneous datasets. This category can further be subdivided into two types based on approach taken i) Hybrid Models - focuses on combining the power of both DL and GBDT or ii)Transformer based models

4.2.1 Hybrid Models

This category can be further categorised into partially differentiable or fully differentiable based on whether all the learn-able parameters are differentiable or not.

1. **Fully differentiable models** Full differentiable models has a great advantage of end to end DL architecture using gradient descent optimizers and thus can have an efficient implementations in modern DL frameworks like PyTorch or TensorFlow using TPU or GPU.

One of the famous hybrid models in this category includes NODE (Neural Oblivion Decision Trees) [Popov et al., 2019] - specially designed for DL on tabular datasets. It's a generalisation of a famous GBDT model CatBoost [Dorogush et al., 2017]. It uses the entmax transformation [Peters et al., 2019] and soft splits to make the entire architecture differentiable. In the main paper, It's been reported that NODE has outperformed several GBDT models like XGBoost. Yet another unique idea was introduced as Net-DNF [Katzir et al., 2021] which uses an inductive bias that considers every decision tree as a disjunctive normal form (DNF) and thus imitating the GBDT models. In the paper author has shown its result for classification tasks where it was close in performance to that of XGBoost (a GBDT model). A very unique idea that has been proposed recently is Network-on-Network (NON) [Luo et al., 2020]. It consists of 3 components: a field wise network (a unique MLP for each columns), an across field network (deciding optimal operations based on input dataset) and finally an operation fusion network. This performed very well compared to MLP but the author hasn't included GBDT models.

2. **Partially differentiable models** These proposed architectures usually aims for combining a non differentiable (decision trees) architecture with a differentiable MLP.

   One of the famous proposal in this category includes DeepGBM - introduced by Microsoft Research [Ke et al., 2019]. This model is a combination of MLP architecture with GBDT in order to provide feature selection capabilities. It includes CatNN for categorical features and GBDT2NN for numerical features. Another remarkable architecture in this category is Boosted Graph Neural Network (BGNN) [Ivanov and Prokhorenkova, 2021] which uses the information from GBDT using graphical neural network.

4.2.2 Transformer based models

The transformer based models have been really famous these days because of their achievements in the field of Natural Language Processing and computer vision. It's basically a DL model that is based on the attention mechanisms [Vaswani et al., 2017] which gives a kind of a certain significance to each part of the input data.

TabNET proposed by Google in 2019 [Arik and Pfister, 2019] has shown a considerable improvement when compared to GBDT as mentioned by author. It basically combines the power of both MLP (to map features to higher dimension, thus effective feature engineering) and GBDT (effective feature selection). It consists of two main blocks (i) Feature transformer responsible for feature engineering (ii) Attention transformer responsible for effective feature selection with the help of a learn-able mask and sparsemax layer. It has several advantages over other DL methods for tabular datasets: (i) handles class imbalance (ii) inbuilt local as well as global explainability (iii) highly robust to large search space of hyper-parameters. This model has opened a new possibility of DL architectures for tabular data. Several architectures based on similar concepts are introduced recently like TabTransformer [Huang et al., 2020], SAINT (Self-Attention and Intersample Attention Transformer) [Somepalli et al., 2021] and ARM-Net [Cai et al., 2021].

4.3 Regularization

As we have discussed above in section 4 that one of the possible reason of ineffectiveness of DL architectures on tabular datasets compared to GBDT is that they are highly sensitive to the

input data. This category of approaches is based on the hypothesis that these high sensitivity issue can be solved by penalising learn-able parameters.

One of the first attempt in this direction to beat GBDT was the Regularization Neural Network [Shavitt and Segal, 2018]. The main idea was to use regularization coefficients that can be trained through gradient descent for each of the learn-able parameters. Recently [Kadra et al., 2021] proposed an approach in which a cocktail of 13 different regularizers are used. Author then selected an optimal subset and specific hyper-parameters for selected regularizers.

5 Stage 2 comparison: So, Have DL architectures really outperformed GBDT?

Many of the recently developed DL architectures claimed to have outperformed GBDT or at least achieved similar performance but most of them have reported their results on different datasets as there's no standard benchmark as in the case of image [Deng et al., 2009] or text [Wang et al., 2018]. Thus, it's quite challenging to have an apple to apple comparison for these models. Furthermore, papers that introduced these new models might not have done same level of hyper-parameter optimisation. Thus it's become unclear that whether the recent DL architectures for tabular data has really surpassed GBDT or not?

In [Shwartz-Ziv and Armon, 2021] author has attempted to solve this problem up to some extent by comparing 3 of the recently developed most promising DL architectures for tabular dataset: TabNET, DNF-Net and NODE with a specific GBDT model XGBoost. This is quite an interesting study where the author had done this comparison on 9 datasets: 3 each from what mentioned in the original paper of TabNET, DNF-Net and NODE and 2 datasets not used in any of them with the same hyper-parameters tuning for all of them. Authors also mentioned that they were able to match the original paper results as well which means their experiment setups are reliable as well. Details of the datasets and detailed comparison is given in Table 4. MSE is used for regression tasks and loglossX100 is used for classification tasks. 2 Best performances are also marked bold. Following observations can be made from the results: (i) These new DL architectures performed best only on the datasets that have been used in original paper. Except for DNF-NET, where XGBoost performed significantly better just once. (ii) On both of the two new datasets, XGBoost gives the best performance than any other DL architectures. (iii) Out of 9 datasets which have been used in some of the original paper, 6 times XGBoost performed best if we will not consider the DL architecture that has been introduced in that paper.

6 Summary

The main question of this survey that whether DL architectures are better than GBDT or not for tabular datasets has been answered in 3 main sections. It starts with a stage 1 comparison where the vanilla DL architecture for tabular datasets (MLP) is compared with GBDT and it has been observed that out of a total of 14 diverse datasets, for 10 of them GBDT has surpassed the MLP. It confirms the fact that there are some issues with DL architecture for tabular datasets - MLP which are discussed in the second section. This will help the reader to determine the possible area of improvement for developing a new DL architecture or pre-processing pipeline for tabular datasets. The motive of this survey is not just to discuss all of recent attempts to surpass GBDT but also to motivate or help the reader to come up with a new improved architecture

Dataset	Features	Classes	Samples	Source	Original Paper
Gesture Phase	32	5	9.8k	OpenML	DNF-Net
Gas Concentrations	129	6	13.9k	OpenML	DNF-Net
Eye Movements	26	3	10.9k	OpenML	DNF-Net
Epsilon	2000	2	500k	PASCAL Challenge 2008	NODE
YearPrediction	90	1	515k	Million Song Dataset	NODE
Microsoft (MSLR)	136	5	964k	MSLR-WEB10K	NODE
Rossmann Store Sales	10	1	1018K	Kaggle	TabNet
Forest Cover Type	54	7	580k	Kaggle	TabNet
Higgs Boson	30	2	800k	Kaggle	TabNet
Shrutime	11	2	10k	Kaggle	New dataset
Blastchar	20	2	7k	Kaggle	New dataset

Model Name	Rossman	CoverType	Higgs	Gas	Eye	Gesture
XGBoost	490.18 ± 1.19	**3.13 ± 0.09**	21.62 ± 0.33	2.18 ± 0.20	**56.07±0.65**	**80.64 ± 0.80**
NODE	**488.59 ± 1.24**	4.15 ± 0.13	**21.19 ± 0.69**	**2.17 ± 0.18**	**68.35 ± 0.66**	92.12 ± 0.82
DNF-Net	503.83 ± 1.41	3.96 ± 0.11	23.68 ± 0.83	**1.44 ±0.09**	68.38 ± 0.65	**86.98 ± 0.74**
TabNet	**485.12±1.93**	**3.01 ± 0.08**	**21.14±0.20**	1.92 ± 0.14	67.13 ± 0.69	96.42 ± 0.87
	TabNET			**DNF-Net**		

	YearPrediction	MSLR	Epsilon	Shrutime	Blastchar
XGBoost	**77.98 ± 0.11**	**55.43±2e-2**	**11.12±3e-2**	**13.82 ± 0.19**	**20.39 ± 0.21**
NODE	**76.39 ± 0.13**	**55.72±3e-2**	**10.39±1e-2**	14.61 ± 0.10	21.40 ± 0.25
DNF-Net	81.21 ± 0.18	56.83±3e-2	12.23±4e-2	16.8 ± 0.09	27.91 ± 0.17
TabNet	83.19 ± 0.19	56.04±1e-2	11.92±3e-2	14.94±, 0.13	23.72 ± 0.19
	NODE			**New Datasets**	

Table 4: i)Datasets description and Test set results for stage 2 comparison. [Shwartz-Ziv and Armon, 2021]

by looking at past attempts. Thus in the last section, instead of discussing all recent attempts, all possible categories of these attempts are discussed like converting a heterogeneous to a homogeneous dataset or building fully or partially differentiable hybrid models, or transformer-based models to have an inbuilt feature selection mechanism inside the DL architecture for the tabular datasets or an efficient cocktail of several regularisation methods. One of the important parts of this last section is our stage 2 comparison. Many recent DL architectures have claimed to surpass the GBDT for the tabular datasets. But due to the lack of a standard benchmark, it is quite hard to have a head-on comparison. An interesting study has been highlighted where the few recent most promising DL models for tabular datasets are compared with GBDT on a set of datasets with the same experimental setups and It has been observed that still for most of the datasets GBDT comes out to be the best or second-best model at least.

7 Conclusion & Future works

To conclude and finally answer the question this survey started with, it can be said DL models for tabular datasets came a long way now compared to the vanilla MLP. But one can still say that GBDTs are state of the art for most of the tabular datasets, obviously, now there exists a few advanced DL architectures (transformer-based models, tabular data converted to image data etc.) which are clearly outperforming GBDT on a certain group of tabular datasets. Considering this survey, future works should consider executing their work on a diverse set of tabular datasets. For future work, instead of looking at GBDT and DL separately, one can also look at the combination of these two as an ensemble model [Shwartz-Ziv and Armon, 2021].

References

[Achille et al., 2019] Achille, A., Paolini, G., and Soatto, S. (2019). Where is the information in a deep neural network? *ArXiv*, abs/1905.12213.

[Amodei et al., 2015] Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J. H., Fan, L., Fougner, C., Han, T., Hannun, A. Y., Jun, B., LeGresley, P., Lin, L., Narang, S., Ng, A. Y., Ozair, S., Prenger, R., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Wang, Y., Wang, Z., Wang, C., Xiao, B., Yogatama, D., Zhan, J., and Zhu, Z. (2015). Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595.

[Arik and Pfister, 2019] Arik, S. Ö. and Pfister, T. (2019). Tabnet: Attentive interpretable tabular learning. *CoRR*, abs/1908.07442.

[Baltrušaitis et al., 2019] Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

[Borisov et al., 2021] Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. (2021). Deep neural networks and tabular data: A survey. *CoRR*, abs/2110.01889.

[Cai et al., 2021] Cai, S., Zheng, K., Chen, G., Jagadish, H. V., Ooi, B. C., and Zhang, M. (2021). Arm-net: Adaptive relation modeling network for structured data. *CoRR*, abs/2107.01830.

[Cattral et al., 2002] Cattral, R., Oppacher, F., and Deugo, D. (2002). Evolutionary data mining with automatic rule generalization.

[Chui,] Chui, M.; Manyika, J. M. M. H. N. C. R. Notes from ai frontier. `https://www.mckinsey.com/~/media/mckinsey/featured%20insights/artificial%20intelligence/notes%20from%20the%20ai%20frontier%20applications%20and%20value%20of%20deep%20learning/notes-from-the-ai-frontier-insights-from-hundreds-of-use-cases-discussion-paper.ashx`.

[Czerniak and Zarzycki, 2003] Czerniak, J. and Zarzycki, H. (2003). Application of rough sets in the presumptive diagnosis of urinary system diseases. In Sołdek, J. and Drobiazgiewicz, L., editors, *Artificial Intelligence and Security in Computing Systems*, pages 41–51, Boston, MA. Springer US.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Dorogush et al., 2017] Dorogush, A. V., Gulin, A., Gusev, G., Kazeev, N., Prokhorenkova, L. O., and Vorobev, A. (2017). Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516.

[Evans and Fisher, 1994] Evans, B. and Fisher, D. (1994). Overcoming process delays with decision tree induction. *IEEE Expert*, 9(1):60–66.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Gorishniy et al., 2021] Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. (2021). Revisiting deep learning models for tabular data. *CoRR*, abs/2106.11959.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Hestness et al., 2017] Hestness, J., Narang, S., Ardalani, N., Diamos, G. F., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. (2017). Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409.

[Huang et al., 2020] Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. S. (2020). Tabtransformer: Tabular data modeling using contextual embeddings. *CoRR*, abs/2012.06678.

[Ivanov and Prokhorenkova, 2021] Ivanov, S. and Prokhorenkova, L. (2021). Boost then convolve: Gradient boosting meets graph neural networks. *CoRR*, abs/2101.08543.

[Kadra et al., 2021] Kadra, A., Lindauer, M., Hutter, F., and Grabocka, J. (2021). Regularization is all you need: Simple neural nets can excel on tabular data. *CoRR*, abs/2106.11189.

[Kaggle,] Kaggle. Kaggle official executive summary. `https://storage.googleapis.com/kaggle-media/surveys/Kaggle's%20State%20of%20Machine%20Learning%20and%20Data%20Science%202021.pdf`.

[Karr et al., 2006] Karr, A. F., Sanil, A. P., and Banks, D. L. (2006). Data quality: A statistical perspective. *Statistical Methodology*, 3(2):137–173.

[Katzir et al., 2021] Katzir, L., Elidan, G., and El-Yaniv, R. (2021). Net-dnf: Effective deep modeling of tabular data. In *ICLR*.

[Ke et al., 2019] Ke, G., Xu, Z., Zhang, J., Bian, J., and Liu, T.-Y. (2019). Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 384–394, New York, NY, USA. Association for Computing Machinery.

[Lai et al., 2015] Lai, S., Xu, L., Liu, K., and Zhao, J. (2015). Recurrent convolutional neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

[Levy et al., 2020] Levy, E., Mathov, Y., Katzir, Z., Shabtai, A., and Elovici, Y. (2020). Not all datasets are born equal: On heterogeneous data and adversarial examples. Workingpaper.

[Luo et al., 2020] Luo, Y., Zhou, H., Tu, W., Chen, Y., Dai, W., and Yang, Q. (2020). Network on network for tabular data classification in real-world applications. *CoRR*, abs/2005.10114.

[Medvedev and D'yakonov, 2020] Medvedev, D. and D'yakonov, A. (2020). New properties of the data distillation method when working with tabular data. *CoRR*, abs/2010.09839.

[Mitchell, 2017] Mitchell, B. (2017). The spatial inductive bias of deep learning.

[Pang et al., 2021] Pang, G., Shen, C., Cao, L., and van den Hengel, A. (2021). Deep learning for anomaly detection. *ACM Computing Surveys (CSUR)*, 54:1 – 38.

[Peters et al., 2019] Peters, B., Niculae, V., and Martins, A. F. T. (2019). Sparse sequence-to-sequence models. *CoRR*, abs/1905.05702.

[Poole et al., 2016] Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

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

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.

[Rahaman et al., 2019] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR.

[Roschewitz et al., 2021] Roschewitz, D., Hartley, M., Corinzia, L., and Jaggi, M. (2021). Ifedavg: Interpretable data-interoperability for federated learning. *CoRR*, abs/2107.06580.

[Sánchez-Morales et al., 2019] Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J., and Figueiras-Vidal, A. R. (2019). Improving deep learning performance with missing values via deletion and compensation. *Neural Computing and Applications*, pages 1–12.

[Shavitt and Segal, 2018] Shavitt, I. and Segal, E. (2018). Regularization learning networks: Deep learning for tabular datasets. In *NeurIPS*.

[Shwartz-Ziv and Armon, 2021] Shwartz-Ziv, R. and Armon, A. (2021). Tabular data: Deep learning is not all you need. *ArXiv*, abs/2106.03253.

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

[Stolfo et al., 1999] Stolfo, S., Fan, W., Lee, W., Prodromidis, A., and Chan, P. (1999). Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the jam project.

[Sun et al., 2019] Sun, B., Yang, L., Zhang, W., Lin, M., Dong, P., Young, C., and Dong, J. (2019). Supertml: Two-dimensional word embedding for the precognition on structured tabular data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2973–2981.

[Szegedy et al., 2014] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

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

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

[Wang et al., 2021] Wang, S., Wu, Z., He, G., Wang, S., Sun, H., and Fan, F. (2021). Semi-supervised classification-aware cross-modal deep adversarial data augmentation. *Future Gener. Comput. Syst.*, 125(C):194–205.

[Xu and Veeramachaneni, 2018] Xu, L. and Veeramachaneni, K. (2018). Synthesizing tabular data using generative adversarial networks. *CoRR*, abs/1811.11264.

[Yoon et al., 2020] Yoon, J., Zhang, Y., Jordon, J., and van der Schaar, M. (2020). Vime: Extending the success of self- and semi-supervised learning to tabular domain. In *NeurIPS*.

[Zhu et al., 2021a] Zhu, J., Shan, Y., Wang, J., Yu, S., Chen, G., and Xuan, Q. (2021a). Deepinsight: Interpretability assisting detection of adversarial samples on graphs. *CoRR*, abs/2106.09501.

[Zhu et al., 2021b] Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y., Doroshow, J., and Stevens, R. (2021b). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports*, 11.

[Zhu et al., 2021c] Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y. A., Doroshow, J. H., and Stevens, R. L. (2021c). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific reports*, 11(1):11325.