TABULAR DATA MODELING VIA CONTEXTUAL EMBEDDINGS

Xin Huang *
Amazon AI
xinxh@amazon.com

Ashish Khetan
Amazon AI
khetan@amazon.com

Milan Cvitkovic
PostEra
mwcvitkovic@gmail.com

Zohar Karnin
Amazon AI
zkarnin@amazon.com

ABSTRACT

We introduce TabTransformer, a new tabular data modeling architecture based on deep self-attention Transformers. Our model works by embedding categorical features in a robust and contextual manual, resulting in better prediction performance. We evaluate TabTransformer for supervised setting through extensive experiments on fifteen publicly available datasets, and conclude that it outperforms the state-of-the-art deep learning methods for tabular data by at least 1.0% on mean AUC. Furthermore, for the semi-supervised setting we develop an unsupervised pre-training and fine-tuning paradigm to learn data-driven contextual embeddings, resulting in an average 2.1% AUC lift over the state-of-the-art methods. Lastly, we demonstrate that the contextual embeddings learned from TabTransformer provide better interpretability, and are highly robust against both missing and noisy data features.

1 INTRODUCTION

Tabular data regression and classification are crucial to many real-world applications such as recommender systems (Cheng et al., 2016), online advertising (Song et al., 2019), and sales forecasting (Pavlyshenko, 2019). Many machine learning competitions such as Kaggle (Kaggle, 2020) and KDD Cup (SIGKDD, 2020) are primarily designed to solve problems in tabular domain, where various machine learning models are built to take each instance (row of tabular data) as input and map it to a target value.

The state-of-the-art for modeling tabular data is tree-based ensemble methods such as the gradient boosted decision trees (GBDT) (Chen & Guestrin, 2016). This differs from modeling image and text data where all the existing competitive models are based on deep learning (Sandler et al., 2018; Devlin et al., 2019). The tree-based ensemble models are accurate, fast to train, and easy to interpret, making them highly favourable among machine learning practitioners. However, their limitations are significant compared with deep learning models: (a) they do not allow efficient end-to-end learning of image/text encoders in presence of multi-modality along with tabular data; (b) they do not fit into the state-of-the-art semi-supervised learning framework due to unreliable probability estimation produced by basic decision tree (Tanha et al., 2017); and (c) they do not enjoy the SoTA deep learning methods (Devlin et al., 2019) to handle missing and noisy data features.

A classical and popular model that is trained using gradient descent and hence allows end-to-end learning of image/text encoders is multi-layer perceptron (MLP). The MLPs usually learn parametric embeddings to encode categorical/continuous data features. But due to their shallow architecture and context-free nature of the learned embeddings, they have the following limitations: (a) neither the model nor the learned embeddings are interpretable; (b) it is not robust against missing and noisy data (Section 3.2); (c) for semi-supervised learning, they do not achieve competitive performance (Section 3.4). Most importantly, the prediction accuracy of MLPs do not match that of tree-based models on most of the datasets (Arik & Pfister, 2019). To bridge this performance gap, researchers

*Corresponding author

have proposed various deep learning models (Arik & Pfister, 2019; Song et al., 2019; Cheng et al., 2016; Guo et al., 2018). Although these deep learning models achieve comparable prediction accuracy, they do not address all the limitations of GBDT and MLP. Furthermore, their comparisons are done in a limited setting of a handful of datasets. In particular, in Section 3.3 we show that when compared to standard GBDT on a large collection of datasets, GBDT perform significantly better than these recent models.

Different from tabular domain, the application of embeddings has been studied extensively in natural language processing. The embedding technique encodes discrete words (a categorical variable) in a dense low dimensional space, beginning from Word2Vec (Rong, 2014) with the context-free word embeddings to BERT (Devlin et al., 2019) which provides the contextual word embeddings. Based on contextual embedding, the self-attention Transformers (Vaswani et al., 2017) has achieved state-of-the-art performance on many NLP tasks. Additionally, the pre-training/fine-tuning paradigm in BERT, which pre-trains the Transformers on a large coprus of unsupervised text and fine-tunes it on downstream tasks with labeled text, has shed light on tabular data modeling in semi-supervised learning.

Motivated by the successful applications of Transformers in NLP, we adapt them in tabular domain. Particularly, TabTransformer modifies a sequence of multi-head attention-based Transformer layers on parametric embeddings to transform them into contextual embeddings, bridging the performance gap between baseline MLP and GBDT models. We investigate the effectiveness and interpretability of the resulting contextual embeddings generated by the Transformers. We find that highly correlated features (including feature pairs in the same column and cross column) result in embedding vectors that are close together in Euclidean distance, whereas no such pattern exists in context-free embeddings learned in a baseline MLP model. We also study the robustness of the TabTransformer against random missing and noisy data. The contextual embeddings make them highly robust in comparison to MLPs. Finally, we exploit the pre-training/fine-tuning methodologies from NLP and propose a semi-supervised learning approach for pre-training TabTransformer using unlabeled data and fine-tuning it on labeled data.

One of the key benefits of our proposed method for semi-supervised learning is the two independent training phases: a costly pre-training phase on unlabeled data and a lightweight fine-tuning phase on labeled data. This differs from many state-of-the-art semi-supervised methods (Chapelle et al., 2009; Oliver et al., 2018; Stretcu et al., 2019) that require a single training job including both the labeled and unlabeled data. The separated training procedure benefits the scenario where the model needs to be pretrained once but fine-tuned multiple times for multiple target variables. This scenario is in fact quite common in the industrial setting as companies tend to have one large dataset (e.g. describing customers/products) and are interested in applying multiple analyses on this data. We summarize our contributions as follows:

1. We propose TabTransformer, an architecture that provides and exploits contextual embeddings of categorical features. We provide extensive experiments showing that TabTransformer is superior to SoTA deep network models.
2. We investigate the resulting contextual embeddings and highlight their interpretability, contrasted to parametric context-free embeddings achieved by existing art.
3. We demonstrate the robustness of TabTransformer against noisy and missing data.
4. We provide and extensively study a two-phase pre-training then fine-tune procedure for tabular data, beating the state-of-the-art performance of semi-supervised learning methods.

2 Architecture and Training Process

The TabTransformer architecture comprises a column embedding layer, a stack of N Transformer layers, and a multi-layer perceptron. Each Transformer layer (Vaswani et al., 2017) consists of a multi-head self-attention layer followed by a position-wise feed-forward layer. The architecture of TabTransformer is shown below in Figure 1.

In our experiments we use standard feature engineering techniques to transform special types such as text, zipcodes, ip addresses etc., into either numeric or categorical features. Although better techniques may exist for handling special data types they are outside the scope of this paper.

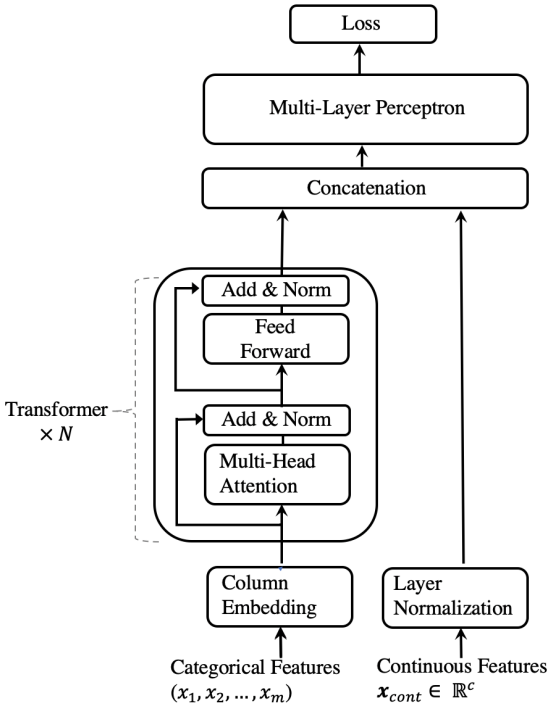

Figure 1: The architecture of TabTransformer.

Let $(\boldsymbol{x}, y)$ denote a features-target pair, where $\boldsymbol{x} \equiv \{\boldsymbol{x}_{\text{cat}}, \boldsymbol{x}_{\text{cont}}\}$ are processed features, and y is target value. The $\boldsymbol{x}_{\text{cat}}$ denotes all the categorical features and $\boldsymbol{x}_{\text{cont}} \in \mathbb{R}^c$ denotes all of the c continuous features. Let $\boldsymbol{x}_{\text{cat}} \equiv \{x_1, x_2, \cdots, x_m\}$ with each x_i being a categorical feature, for $i \in \{1, \cdots, m\}$. We embed each of the x_i categorical features into a parametric embedding of dimension d using *Column embedding*, which is explained below in detail. Let $\boldsymbol{e}_{\phi_i}(x_i) \in \mathbb{R}^d$ for $i \in \{1, \cdots, m\}$ be the embedding of the x_i feature, and $\boldsymbol{E}_\phi(\boldsymbol{x}_{\text{cat}}) = \{\boldsymbol{e}_{\phi_1}(x_1), \cdots, \boldsymbol{e}_{\phi_m}(x_m)\}$ be the set of embeddings for all the categorical features.

Next, these parametric embeddings $\boldsymbol{E}_\phi(\boldsymbol{x}_{\text{cat}})$ are passed through N Transformer layers. Each parametric embedding is transformed into contextual embedding when outputted from the top layer Transformer, through successive aggregation of context from other embeddings. We denote the sequence of N Transformer layers as a function f_θ. The function f_θ operates on parametric embeddings $\{\boldsymbol{e}_{\phi_1}(x_1), \cdots, \boldsymbol{e}_{\phi_m}(x_m)\}$ and returns the corresponding contextual embeddings $\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_m\}$ where $\boldsymbol{h}_i \in \mathbb{R}^d$ for $i \in \{1, \cdots, m\}$. The contextual embeddings $\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_m\}$ are concatenated along with the continuous features $\boldsymbol{x}_{\text{cont}}$ to form a vector of dimension $(d \times m + c)$. This vector is inputted to an MLP, denoted by $g_{\boldsymbol{\psi}}$, to predict the target y. Let H be the cross-entropy for classification tasks and mean square error for regression tasks. We minimize the following loss function $\mathcal{L}(\boldsymbol{x}, y)$ to learn all the TabTransformer parameters in an end-to-end learning by the first-order gradient methods. The TabTransformer parameters include $\boldsymbol{\phi}$ for column embedding, $\boldsymbol{\theta}$ for Transformer layers, and $\boldsymbol{\psi}$ for the top MLP layer.

$$\mathcal{L}(\boldsymbol{x}, y) \equiv H(g_{\boldsymbol{\psi}}(f_{\boldsymbol{\theta}}(\boldsymbol{E}_\phi(\boldsymbol{x}_{\text{cat}})), \boldsymbol{x}_{\text{cont}}), y)\,. \tag{1}$$

Below, we explain the column embedding.

Column embedding. For each categorical feature (column) i, we have an embedding lookup table $\boldsymbol{e}_{\phi_i}(.)$, for $i \in \{1, 2, ..., m\}$. For ith feature with d_i classes, the embedding table $\boldsymbol{e}_{\phi_i}(.)$ has $(d_i + 1)$ embeddings where the additional embedding corresponds to a missing value. The embedding for the encoded value $x_i = j \in [0, 1, 2, .., d_i]$ is $\boldsymbol{e}_{\phi_i}(j) = [\boldsymbol{c}_{\phi_i}, \boldsymbol{w}_{\phi_{ij}}]$, where $\boldsymbol{c}_{\phi_i} \in \mathbb{R}^\ell$ and $\boldsymbol{w}_{\phi_{ij}} \in \mathbb{R}^{d-\ell}$. The column-specific and unique identifier $\boldsymbol{c}_{\phi_i} \in \mathbb{R}^\ell$ distinguishes the classes in column i from those in the other columns. The dimension of $\boldsymbol{c}_{\phi_i}$, ℓ, is a hyper-parameter.

The use of unique identifier is innovative and particularly designed for tabular data. Rather in NLP, embeddings are element-wisely added with the positional encoding of the word in the sentence. Since, in tabular data, there is no ordering of the features, we do not use positional encodings. The strategies include both different choices for ℓ, d and element-wise adding the unique identifier and feature-value specific embeddings rather than concatenating them.

Pre-training the Embeddings. The contextual embeddings explained above are learned in end-to-end supervised training using labeled examples. For a scenario, when there are a few labeled examples and a large number of unlabeled ones, we introduce a pre-training procedure to train the Transformer layers using unlabeled data. This is followed by fine-tuning of the pre-trained Transformer layers along with the top MLP layer using the labeled data. For fine-tuning, we use the supervised loss defined in Equation 1.

We explore two different types of pre-training procedures, the masked language modeling (MLM) (Devlin et al., 2019) and the replaced token detection (RTD) (Clark et al., 2020). Given an input $\boldsymbol{x}_{\text{cat}} = \{x_1, x_2, ..., x_m\}$, MLM randomly selects $k\%$ features from index 1 to m and masks them as missing. The Transformer layers along with the column embeddings are trained by minimizing cross-entropy loss of a multi-class classifier, which predicts the original features of the masked features using contextual embeddings outputted from the top-layer Transformer.

Instead of masking features, RTD replaces the original feature by a random value of that feature. Here, the loss is minimized for a binary classifier that predicts whether or not the feature has been replaced. To compute the replacement value, the original RTD in Clark et al. (2020) uses an encoder network to sample a subset of features. The reason to use an encoder network is that there are tens of thousands of tokens in language data and a uniformly random token can be easily detected. In contrast, we use uniformly random values to replace tabular features because (a) the number of classes within each categorical feature is typically limited; (b) a different binary classifier is defined for each column rather than a shared one, as each column has its own embedding lookup table. We name the two pre-training methods as TabTransformer-MLM and TabTransformer-RTD. In our experiments, the replacement value k is set to 30.

3 Experiments

Data. We evaluate TabTransformer and baseline models on 15 publicly available binary classification datasets from the UCI repository (Dua & Graff, 2017), the AutoML Challenge (Guyon et al., 2019), and Kaggle (Kaggle, 2020) for both supervised and semi-supervised learning. Each dataset is divided into five cross-validation splits. The training/validation/testing proportion of the data for each split are $65/15/20\%$. The number of categorical features across dataset ranges from 2 to 136. In the semi-supervised experiments, for each dataset and split, p observations in the training data are uniformly sampled as the labeled data with a fixed random seed and the remaining training data are set as unlabeled set. The value of p is chosen as 50, 200, and 500, corresponding to 3 different scenarios. In the supervised experiments, each training dataset is fully labeled.

Setup. For TabTransformer, the hidden (embedding) dimension, the number of layers and the number of attention heads are fixed to 32, 6, and 8 respectively; these parameters are pre-selected by hyperparameter optimization (HPO) on a small number of datasets. The MLP layer sizes are set to $\{4 \times l, 2 \times l\}$, where l is the size of its input. Each competitor model is given 20 HPO rounds for each cross-validation split. For evaluation metrics, we use the Area under the curve (AUC) (Bradley, 1997). Note, the pre-training is only applied in semi-supervised scenario. We do not find much benefit in using it when the entire data is labeled. Its benefit is evident when there is a large number of unlabeled examples and a few labeled examples. Since in this scenario the pre-training provides a representation of the data that could not have been learned based only on the labeled examples.

3.1 The Effectiveness of the Transformer Layers

First, a comparison between TabTransformers and the baseline MLP is conducted in a supervised learning scenario. We remove the Transformer layers $f_{\boldsymbol{\theta}}$ from the architecture, fix the rest of the components, and compare it with the original TabTransformer. The model without the Transformer layers is equivalently an MLP. The dimension of embeddings d for categorical features is set as 32 for both models. The comparison results over 15 datasets are presented in Table 1. The TabTransformer

Table 1: Comparison between TabTransfomers and the baseline MLP. The evaluation metric is AUC in percentage.

Dataset	Baseline MLP	TabTransformer	Gain (%)
albert	74.0	75.7	**1.7**
1995_income	90.5	90.6	**0.1**
dota2games	63.1	63.3	**0.2**
hcdr_main	74.3	75.1	**0.8**
adult	72.5	73.7	**1.2**
bank_marketing	92.9	93.4	**0.5**
blastchar	83.9	83.5	-0.4
insurance_co	69.7	74.4	**4.7**
jasmine	85.1	85.3	**0.2**
online_shoppers	91.9	92.7	**0.8**
philippine	82.1	83.4	**1.3**
qsar_bio	91.0	91.8	**0.8**
seismicbumps	73.5	75.1	**1.6**
shrutime	84.6	85.6	**1.0**
spambase	98.4	98.5	**0.1**

with the Transformer layers outperforms the baseline MLP on 14 out of 15 datasets with an average 1.0% gain in AUC.

Next, we take contextual embeddings from different layers of the Transformer and compute a t-SNE plot (Maaten & Hinton, 2008) to visualize their similarity in function space. More precisely, for each dataset we take its test data, pass their categorical features into a trained TabTransformer, and extract all contextual embeddings (across all columns) from a certain layer of the Transformer. The t-SNE algorithm is then used to reduce each embedding to a 2D point in the t-SNE plot. Figure 2 (left) shows the 2D visualization of embeddings from the last layer of the Transformer for dataset *bank_marketing*. Each marker in the plot represents an average of 2D points over the test data points for a certain class. We can see that semantically similar classes are close with each other and form clusters (annotated by a set of labels) in the embedding space. For example, all of the client-based features (color markers) such as job, education level and martial status stay close in the center and non-client based features (gray markers) such as month (last contact month of the year), day (last contact day of the week) lie outside the central area; in the bottom cluster the embedding of owning a housing loan stays close with that of being default; over the left cluster, embeddings of being a student, martial status as single, not having a housing loan, and education level as tertiary get together; and in the right cluster, education levels are closely associated with the occupation types (Torpey & Watson, 2014). In Figure 2, the center and right plots are t-SNE plots of embeddings before being passed through the Transformer and the context-free embeddings from MLP, respectively. For the embeddings before being passed into the Transformer, it starts to distinguish the non-client based features (gray markers) from the client-based features (color markers). For the embeddings from MLP, we do not observe such pattern and many categorical features which are not semantically similar are grouped together, as indicated by the annotation in the plot. We also evaluate the effectiveness of TabTransformer by fitting embeddings extracted from different layers into a linear model.

3.2 The Robustness of TabTransformer

We demonstrate the robustness of TabTransformer on the noisy data and data with missing values, against the baseline MLP. We consider these two scenarios only on categorical features to specifically prove the robustness of contextual embeddings from the Transformer layers.

Noisy Data. On the test examples, we firstly contaminate them by replacing a certain number of values by randomly generated ones from the corresponding columns (features). Next, the noisy data are passed into a trained TabTransformer to compute a prediction AUC score. Results on a set of 3 datasets are presented in Figure 3. As the noisy rate increases, TabTransformer performs better

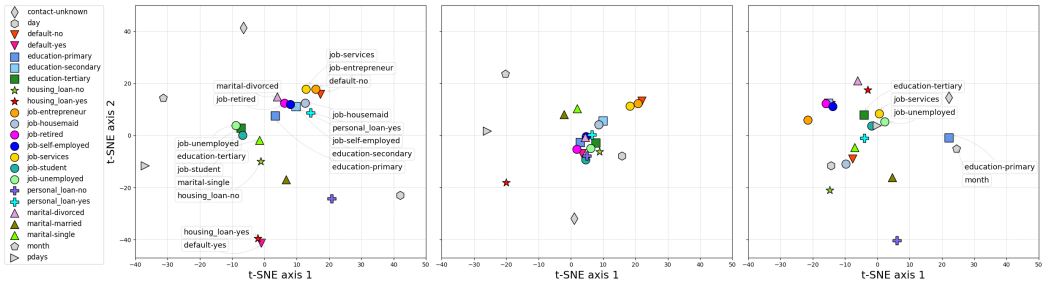

Figure 2: t-SNE plots of learned embeddings for categorical features on dataset *BankMarketing*. **Left**: TabTransformer – the embeddings generated from the last layer of the attention-based Transformer. **Center**: TabTransformer – the embeddings before being passed into the attention-based Transformer. **Right**: The embeddings learned from MLP.

in prediction accuracy and thus is more robust than MLP. In particular notice the *Blastchar* dataset where the performance is near identical with no noise, yet as the noise increases, TabTransformer becomes significantly more performant compared to the baseline. We conjecture that the robustness comes from the contextual property of the embeddings. Despite a feature being noisy, it draws information from the correct features allowing for a certain amount of correction.

Data with Missing Values. Similarly, on the test data we artificially select a number of values to be missing and send the data with missing values to a trained TabTransformer to compute the prediction score. Figure 4 shows the same patterns of the noisy data case, i.e. that TabTransformer shows better stability than MLP in handling missing values.

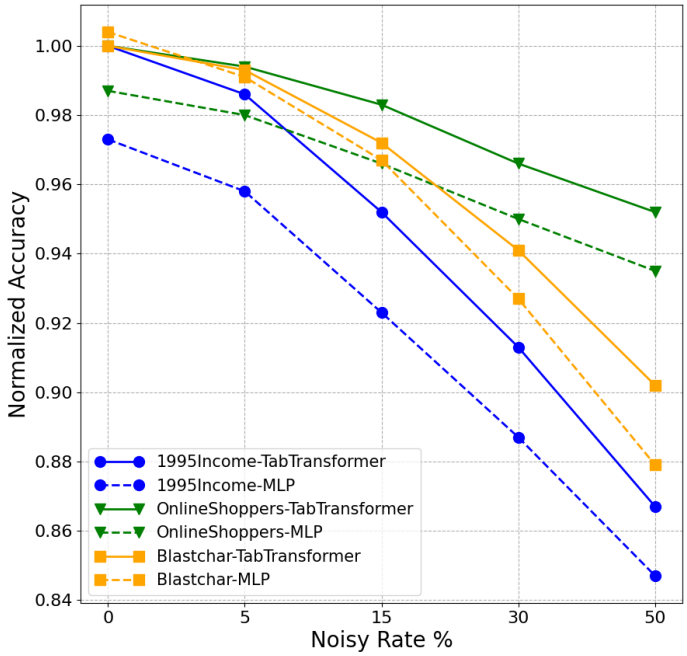

Figure 3: Performance of TabTransformer and MLP with noisy data. For each dataset, each prediction score is normalized by the score of TabTransformer at 0 noise.

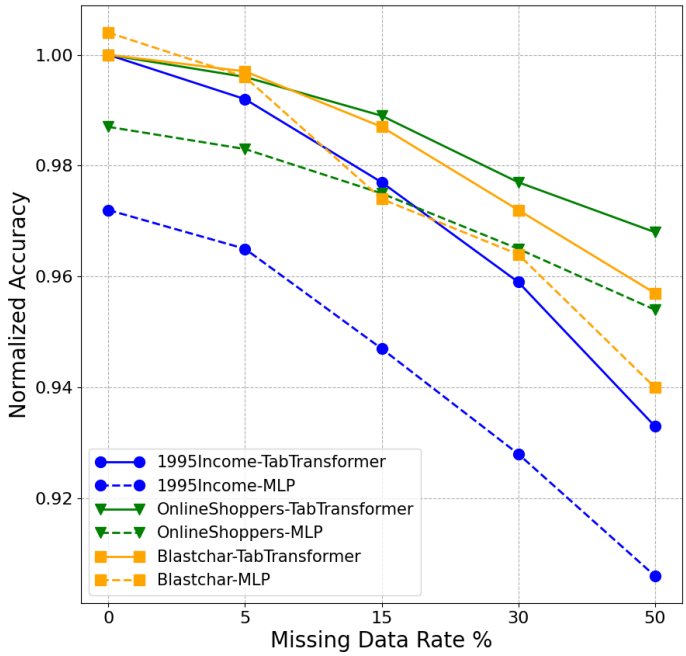

Figure 4: Performance of TabTransformer and MLP under missing data scenario. For each dataset, each prediction score is normalized by the score of TabTransformer trained without missing values.

Table 2: Model performance in supervised learning. The evaluation metric is mean $\pm$ standard deviation of AUC score over the 15 datasets for each model. Larger the number, better the result. The top 2 numbers are bold.

Model Name	**Mean AUC (%)**
TabTransformer	$\mathbf{82.8} \pm 0.4$
MLP	81.8 ± 0.4
GBDT	$\mathbf{82.9} \pm 0.4$
Sparse MLP	81.4 ± 0.4
Logistic Regression	80.4 ± 0.4
TabNet	77.1 ± 0.5
VIB	80.5 ± 0.4

3.3 Supervised Learning

Here we compare the performance of TabTransformer against following four categories of methods: (a) Logistic regression and GBDT; (b) MLP and a sparse MLP following Morcos et al. (2019); (c) TabNet model of Arik & Pfister (2019); and (d) the Variational Information Bottleneck model (VIB) of Alemi et al. (2017).

Results are summarized in Table 2. TabTransformer, MLP, and GBDT are the top 3 performers. The TabTransformer outperforms the baseline MLP with an average 1.0% gain and perform comparable with the GBDT. Furthermore, the TabTransformer is significantly better than TabNet and VIB, the recent deep networks for tabular data.

3.4 Semi-supervised Learning

We evaluate the TabTransformer under the semi-supervised learning scenario where few labeled training examples are available together with a significant number of unlabeled samples. Specifically, we compare our pretrained and then fine-tuned TabTransformer-RTD/MLM against following

Table 3: Semi-supervised learning results for 6 datasets with more than 30K data points, for different number of labeled data points. Evaluation metrics are mean AUC in percentage. Larger the number, better the result.

# Labeled data	**50**	**200**	**500**
TabTransformer-RTD	66.6 ± 0.6	70.9 ± 0.6	$\mathbf{73.1} \pm 0.6$
TabTransformer-MLM	$\mathbf{66.8} \pm 0.6$	$\mathbf{71.0} \pm 0.6$	72.9 ± 0.6
MLP (ER)	65.6 ± 0.6	69.0 ± 0.6	71.0 ± 0.6
MLP (PL)	65.4 ± 0.6	68.8 ± 0.6	71.0 ± 0.6
TabTransformer (ER)	62.7 ± 0.6	67.1 ± 0.6	69.3 ± 0.6
TabTransformer (PL)	63.6 ± 0.6	67.3 ± 0.7	69.3 ± 0.6
MLP (DAE)	65.2 ± 0.5	68.5 ± 0.6	71.0 ± 0.6
GBDT (PL)	56.5 ± 0.5	63.1 ± 0.6	66.5 ± 0.7

semi-supervised models: (a) Entropy Regularization (ER) (Grandvalet & Bengio, 2006) combined with MLP and TabTransformer; (b) Pseudo Labeling (PL) (Lee, 2013) combined with MLP, TabTransformer, and GBDT (Jain, 2017); (c) MLP (DAE): an unsupervised pre-training method designed for deep models on tabular data – the swap noise Denoising AutoEncoder (Jahrer, 2018).

The pre-training models TabTransformer-MLM, TabTransformer-RTD and MLP (DAE) are firstly pretrained on the entire unlabeled training data and then fine-tuned on labeled data. The semi-supervised learning methods, Pseudo Labeling and Entropy Regularization, are trained on the mix of labeled and unlabeled training data. To better present results, we split the set of 15 datasets into two subsets. The first set includes 6 datasets with more than 30K data points and the second set includes remaining 9 datasets.

The results are presented in Table 3 and Table 4. When the number of unlabeled data is large, Table 3 shows that our TabTransformer-RTD and TabTransformer-MLM significantly outperform all the other competitors. Particularly, TabTransformer-RTD/MLM improves over all the other competitors by at least 1.2%, 2.0% and 2.1% on mean AUC for the scenario of 50, 200, and 500 labeled data points respectively. The Transformer-based semi-supervised learning methods TabTransformer (ER), TabTransformer (PL), and the tree-based semi-supervised learning method GBDT (PL) perform worse than the average of all the models. When the number of unlabeled data becomes smaller, as shown in Table 4, TabTransformer-RTD still outperforms most of its competitors but with a marginal improvement.

Furthermore, we observe that when the number of unlabeled data is small as shown in Table 4, TabTransformer-RTD performs better than TabTransformer-MLM, thanks to its easier pre-training task (a binary classification) than that of MLM (a multi-class classification). This aligns with the finding of the ELECTRA paper (Clark et al., 2020). In Table 4, with only 50 labeled data points, MLP (ER) and MLP (PL) beat our TabTransformer-RTD/MLM. This can be attributed to that there is room to improve in our fine-tuning procedure. Particularly, our approach allows to obtain informative embeddings but does not allow the weights of the classifier itself to be trained with unlabelled data. Since this issue does not occur for ER and PL, they obtain an advantage in extremely small labelled set. We point out however that this only means that the methods are complementary and a possible follow up could combine the best of all approaches.

Both evaluation results, Table 3 and Table 4, show that our TabTransformer-RTD and Transformers-MLM models are promising in extracting useful information from unlabeled data to help supervised training, and are particularly useful when the size of unlabeled data is large.

4 Related Work

For supervised learning, standard MLPs have been applied to tabular data for many years (De Brébisson et al., 2015). For deep models designed specifically for tabular data, there are deep versions of factorization machines (Guo et al., 2018; Xiao et al., 2017), deep MLPs-based methods (Wang et al., 2017; Cheng et al., 2016; Cortes et al., 2016), Transformers-based methods (Song et al., 2019; Li et al., 2020; Sun et al., 2019), and deep versions of decision-tree-based algorithms

Table 4: Semi-supervised learning results for 9 datasets with less than 30K data points, for different number of labeled data points. Evaluation metrics are mean AUC in percentage. Larger the number, better the result.

# Labeled data	**50**	**200**	**500**
TabTransformer-RTD	78.6 ± 0.6	$\mathbf{81.6} \pm 0.5$	$\mathbf{83.4} \pm 0.5$
TabTransformer-MLM	78.5 ± 0.6	81.0 ± 0.6	82.4 ± 0.5
MLP (ER)	$\mathbf{79.4} \pm 0.6$	81.1 ± 0.6	82.3 ± 0.6
MLP (PL)	79.1 ± 0.6	81.1 ± 0.6	82.0 ± 0.6
TabTransformer (ER)	77.9 ± 0.6	81.2 ± 0.6	82.1 ± 0.6
TabTransformer (PL)	77.8 ± 0.6	81.0 ± 0.6	82.1 ± 0.6
MLP (DAE)	78.5 ± 0.7	80.7 ± 0.6	82.2 ± 0.6
GBDT (PL)	73.4 ± 0.7	78.8 ± 0.6	81.3 ± 0.6

(Ke et al., 2019; Yang et al., 2018). In particular, Song et al. (2019) apply one layer of multi-head attention on embeddings to learn higher order features. The higher order features are concatenated and inputted to a fully connected layer to make the final prediction. Li et al. (2020) use self-attention layers and track the attention scores to obtain feature importance scores. Sun et al. (2019) combine the Factorization Machine model with transformer mechanism. All 3 papers are focused on recommendation systems with input data being high dimensional and extremely sparse, which makes it hard to have a clear comparison with this paper. Recent TabNet (Arik & Pfister, 2019) is designed on the sparse feature interaction of tabular data, and has a very different mechanism than our self-attention based one. There are a few works focusing on database oriented task in tabular domain, such as column categorization (Chen et al., 2019), entity linking (Luo et al., 2018), table layout identification (Habibi et al., 2020), and table augmentation (Deng et al., 2019). However, these tasks are fundamentally different from typical ML classification problem. They do not classify individual rows of a table, but properties of the table itself, i.e., meta-data. For this reason we do not elaborate the details of these papers nor compare our results with theirs.

For semi-supervised learning, Izmailov et al. (2019) give a semi-supervised method based on density estimation and evaluate their approach on tabular data. *Pseudo labeling* (Lee, 2013) is an efficient and popular baseline method. The pseudo labeling uses the current network to infer pseudo-labels of unlabeled examples, by choosing the most confident class. These pseudo-labels are treated like human-provided labels in the cross entropy loss. *Label propagation* (Zhu & Ghahramani, 2002; Iscen et al., 2019) is a similar approach where a node's labels propagate to all nodes according to their proximity, and are used by the training model as if they were the true labels. Another standard method in semi-supervised learning is *entropy regularization* (Grandvalet & Bengio, 2005; Sajjadi et al., 2016). It adds average per-sample entropy for the unlabeled examples to the original loss function for the labeled examples. Additionally, a classical approach of semi-supervised learning is co-training (Nigam & Ghani, 2000). However, the recent approaches - entropy regularization and pseudo labeling - are typically better and more popular. A succinct review of semi-supervised learning methods in general can be found in Oliver et al. (2019); Chappelle et al. (2010).

5 Conclusion

We proposed TabTransformer, a novel deep tabular data modeling architecture for supervised and semi-supervised learning. Extensive experiments show that TabTransformer significantly outperforms recent deep networks while matching the performance of GBDT. In addition, we study a two-phase pre-training/fine-tune paradigm for tabular data, beating the state-of-the-art semi-supervised learning methods. For future work, it would be interesting to investigate them in detail.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *International Conference on Learning Representations*, abs/1612.00410, 2017. URL https://arxiv.org/abs/1612.00410.

Sercan O Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442*, 2019. URL https://arxiv.org/abs/1908.07442.

Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning). *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

O Chappelle, B Schölkopf, and A Zien. Semi-supervised learning. adaptive computation and machine learning, 2010.

Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. Learning semantic annotations for tabular data. *arXiv preprint arXiv:1906.00781*, 2019.

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*, pp. 785–794, 2016.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10, 2016.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Corinna Cortes, Xavi Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Adanet: Adaptive structural learning of artificial neural networks, 2016.

Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. In *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge - Volume 1526*, ECMLPKDDDC'15, pp. 40–51, Aachen, DEU, 2015. CEUR-WS.org.

Li Deng, Shuo Zhang, and Krisztian Balog. Table2vec: neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1029–1032, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.

Yves Grandvalet and Yoshua Bengio. Entropy regularization. *Semi-supervised learning*, pp. 151–168, 2006.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He, and Zhenhua Dong. DeepFM: An End-to-End Wide & Deep Learning Framework for CTR Prediction. *arXiv:1804.04950 [cs, stat]*, May 2018. URL http://arxiv.org/abs/1804.04950. arXiv: 1804.04950.

Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michéle Sebag, Alexander Statnikov, Wei-Wei Tu, and Evelyne Viegas. Analysis of the automl challenge series 2015-2018. In *AutoML*, Springer series on Challenges in Machine Learning, 2019. URL https://www.automl.org/wp-content/uploads/2018/09/chapter10-challenge.pdf.

Maryam Habibi, Johannes Starlinger, and Ulf Leser. Deeptable: a permutation invariant neural network for table orientation classification. *Data Mining and Knowledge Discovery*, 34(6):1963–1983, 2020.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-Supervised Learning with Normalizing Flows. *arXiv:1912.13025 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.13025. arXiv: 1912.13025.

Michael Jahrer. Porto Seguro's Safe Driver Prediction, 2018. URL https://kaggle.com/c/porto-seguro-safe-driver-prediction.

Shubham Jain. Introduction to pseudo-labelling : A semi-supervised learning technique. https://www.analyticsvidhya.com/blog/2017/09/pseudo-labelling-semi-supervised-learning-technique/, 2017.

Inc. Kaggle. 2020 kaggle machine learning & data science survey, 2020. URL https://www.kaggle.com/c/kaggle-survey-2020.

Guolin Ke, Jia Zhang, Zhenhui Xu, Jiang Bian, and Tie-Yan Liu. TabNN: A universal neural network solution for tabular data, 2019. URL https://openreview.net/forum?id=r1eJssCqY7.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 2, 2013.

Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. Interpretable Click-Through Rate Prediction through Hierarchical Attention. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 313–321, Houston TX USA, January 2020. ACM. ISBN 978-1-4503-6822-3. doi: 10.1145/3336191.3371785. URL http://dl.acm.org/doi/10.1145/3336191.3371785.

Xusheng Luo, Kangqi Luo, Xianyang Chen, and Kenny Zhu. Cross-lingual entity linking for web tables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *arXiv:1906.02773 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1906.02773. arXiv: 1906.02773.

Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pp. 86–93, 2000.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. *arXiv:1804.09170 [cs, stat]*, June 2019. URL http://arxiv.org/abs/1804.09170. arXiv: 1804.09170.

Bohdan M Pavlyshenko. Machine-learning models for sales time series forecasting. *Data*, 4(1):15, 2019.

Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pp. 1163–1171, 2016.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

SIGKDD, 2020. URL https://www.kdd.org/kdd-cup.

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. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management - CIKM '19*, pp. 1161–1170, 2019. doi: 10.1145/3357384.3357925. URL http://arxiv.org/abs/1810.11921. arXiv: 1810.11921.

Otilia Stretcu, Krishnamurthy Viswanathan, Dana Movshovitz-Attias, Emmanouil Platanios, Sujith Ravi, and Andrew Tomkins. Graph Agreement Models for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32*, pp. 8713–8723. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9076-graph-agreement-models-for-semi-supervised-learning.pdf.

Qiang Sun, Zhinan Cheng, Yanwei Fu, Wenxuan Wang, Yu-Gang Jiang, and Xiangyang Xue. DeepEnFM: Deep neural networks with Encoder enhanced Factorization Machine. September 2019. URL https://openreview.net/forum?id=SJlyta4YPS.

Jafar Tanha, Maarten Someren, and Hamideh Afsarmanesh. Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8:355–370, 2017.

Elka Torpey and Audrey Watson. *Education level and jobs: Opportunities by state*, 2014. URL https://www.bls.gov/careeroutlook/2014/article/education-level-and-jobs.htm.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *ADKDD@KDD*, 2017.

Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 3119–3125, Melbourne, Australia, August 2017. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3. doi: 10.24963/ijcai.2017/435. URL https://www.ijcai.org/proceedings/2017/435.

Yongxin Yang, Irene Garcia Morillo, and Timothy M Hospedales. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*, 2018.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.