TabLog: Test-Time Adaptation for Tabular Data Using Logic Rules

Weijieying Ren [1] **Xiaoting Li** [2] **Huiyuan Chen** [2] **Vineeth Rakesh** [2] **Zhuoyi Wang** [2] **Mahashweta Das** [2]
Vasant Honavar [1 3]

Abstract

We consider the problem of test-time adaptation of predictive models trained on tabular data. Effective solution of this problem requires adaptation of predictive models trained on the source domain to a target domain, using only unlabeled target domain data, without access to source domain data. Existing test-time adaptation methods for tabular data have difficulty coping with the heterogeneous features and their complex dependencies inherent in tabular data. To overcome these limitations, we consider test-time adaptation in the setting wherein the *logical structure* of the rules is assumed to remain invariant despite distribution shift between source and target domains whereas the numerical parameters associated with the rules and the weights assigned to them can vary to accommodate distribution shift. TabLog discretizes numerical features, models dependencies between heterogeneous features, introduces a novel contrastive loss for coping with distribution shift, and presents an end-to-end framework for efficient training and test-time adaptation by taking advantage of a logical neural network representation of a rule ensemble. We present results of experiments using several benchmark data sets that demonstrate TabLog is competitive with or improves upon the state-of-the-art methods for test-time adaptation of predictive models trained on tabular data. Our code is available at `https://github.com/WeijieyingRen/TabLog`.

[1]Artificial Intelligence Research Laboratory, Center for Artificial Intelligence Foundations and Scientific Applications, Institute for Computational and Data Sciences, College of Information Sciences and Technology, The Pennsylvania State University, PA, United States. [2]Visa Research, CA, United States [3]College of Information Sciences and Technology, The Pennsylvania State University, PA, United States. Correspondence to: Vasant Honavar <vuh14@psu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Tabular data are ubiquitous in many applications including clinical diagnosis (Somepalli et al., 2022; Arik & Pfister, 2021), climate modeling (Shi et al., 2021), e-commerce (Gardner et al., 2023), among others. Although a variety of methods exist for learning predictive models from tabular data, they suffer from substantial drops in accuracy when the data distribution during model deployment, i.e., the target domain distribution, is significantly different from that encountered during model training, i.e., the source domain distribution (Quiñonero-Candela et al., 2022; Bahri et al., 2021; Wang & Chen, 2022; Gardner et al., 2023). Hence, there is much work on domain adaptation methods that aim to transfer knowledge under distribution shift from a source domain to a target domain (Kundu et al., 2020). Of particular interest is the problem of test-time or source-free (Liang et al., 2023; Fang et al., 2024) domain adaptation where the source data is unavailable during adaptation to the target domain.

Despite notable advances in test-time adaptation (Liang et al., 2021; Wang et al., 2020; Niu et al., 2022; Wang et al., 2023a; Liang et al., 2020) (see (Liang et al., 2023; Fang et al., 2024) for surveys), there has been limited progress on **t**est-**t**ime **a**daptation of predictive models trained on **tab**ular data (TabTTA). Tabular data present several challenges to existing domain adaptation methods: (i) They are characterized by heterogeneous (boolean, categorical, numeric) features, with different underlying distributions (Bahri et al., 2021; Chen & Guestrin, 2016). (ii) The dependencies between features are often complex, and a priori unknown (Shi et al., 2021; Gardner et al., 2023). Hence, some of the state-of-the-art domain adaptation methods, e.g., those that rely on feature alignment (Chen et al., 2019) or feature transfer (Bengio, 2012; Wang & Chen, 2022) fail on tabular data. Effective solution of the TAbTTA problem have to come to terms with two questions: (i) What information can be reasonably transferred from the source domain to the target domain and (ii) How to perform such transfer.

Motivated by the above considerations and the needs of practical applications, we introduce **TabLog**, a novel test-time adaptation technique for **tab**ular data. TabLog learns an ensemble of rules for classifying tabular data in the

source domain. Each rule is of the form `Antecedent` $\rightarrow$ `Consequent`. The `Antecedent` is a logical formula made of atoms (logical propositions that evaluate to True or False) and logical connectives (Riegel et al., 2020) (e.g., $\wedge$, $\vee$) and the `Consequent` is a class label. It assumes that the **log**ical structure of the rules learned from the labeled data in the source domain are unaffected by the distribution shift but the numeric parameters associated with the rules and their relative weights may have to change to accommodate distribution shift. To cope with heterogeneous features, we map each numerical feature into discretized bins with learnable thresholds. These discretized bins and one-hot categorical features are used as atoms in constructing rules. We use logical connectives to model interactions between features. The resulting propositional rules offer a compact representation of learned knowledge and support transfer its logical structure from the source domain to the target domain. To improve test-time adaptation on the target domain, we introduce a new binning-informed contrastive loss function that helps the classifier adapt to covariate shift in the target domain relative to the source domain. The proposed loss function is based on the heuristic that the data representation and the learned rules should be robust enough to accommodate minor variations in the input.

The main contributions of this paper include: (1) A novel general framework for test-time domain adaptation for predictive models learned from tabular data under the assumption that the *logical structure* of the rules learned from the source domain remain invariant under distribution shift; (2) A novel approach to learning an ensemble of rules for classifying tabular data with heterogeneous features, performs test-time adaptation of the parameters of the resulting rule ensemble using unlabeled data from the target domain using a novel contrastive loss function while preserving their logical structure learned from labeled data from the source domain; (3) TabLog, an end-to-end pipeline for efficient training and test-time adaptation of a weighted rule ensamble for tabular data classification by taking advantage of a logical neural network representation of the rule ensemble; (4) Results of extensive experiments on several benchmark tabular data sets that demonstrate that TabLog outperforms the state-of-the-art methods for test-time adaptation of predictive models learned from tabular data.

2. Related Work

Test time adaptation. Most existing approaches to test-time adaptation (TTA) aim to address covariate shift, that is, changes in the marginal distribution, i.e., $P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$, while the conditional distributions remain unchanged, i.e., $P(y_S|\mathbf{X}_S) = P(y_T|\mathbf{X}_T)$. One class of TTA methods include those that exploit class prototypes (Chen et al., 2022; Jang et al., 2022), pseudo labels (Goyal et al., 2022; Liang et al., 2020; Ren et al., 2022), or invariance to augmentation (Gong et al., 2022). A second class of TTA methods make use of self-supervised learning (SSL) on *pretext* or *auxiliary* tasks, e.g., rotation prediction (Liang et al., 2021; Sun et al., 2020) and masked autoencoders (Gandelsman et al., 2022; Ren et al., 2023). A third class of approaches is Test-Time Training (TTT) (reviewed in (Liang et al., 2023)) wherein the source model is trained simultaneously on the primary task and an auxiliary SSL task. During TTA, the source model parameters are updated on the unlabeled target data, without accessing the source domain data.

Domain Adaptation with Tabular Data. Despite much recent work on deep learning methods for tabular data (Bahri et al., 2021; Shi et al., 2021; Yoon et al., 2020; Zhang et al., 2023; Arik & Pfister, 2021; Gardner et al., 2023; Wang et al., 2023a), TTA of such models has received limited attention in the literature. One approach to TTA with such models uses a novel pretext task of estimating mask vectors from corrupted tabular data in addition to the reconstruction pretext task for SSL (Yoon et al., 2020). The second approaches incorporate contrastive loss to enhance the robustness of the learned representation (Bahri et al., 2021; Somepalli et al., 2022; Yun et al., 2019; Wang & Sun, 2022; Ucar et al., 2021). Unlike these methods which seek to learn a distribution shift invariant representation for tabular data, TabLog focuses on rule-based classifiers whose logical structure remains invariant to distribution shift, but its parameters are adapted using unlabeled target domain data. Besides, we find binning can help to design a SSL task.

Differentiable Logics. Differentiable logics, also known as real-valued logics or infinite-valued logic (De Geus & Cohen, 1985; Nilsson, 1986), including continuous fuzzy logics (Cignoli et al., 2000) extend Boolean logic by relaxing discrete truth values in $\{0, 1\}$ to truth degrees in [0, 1], and Boolean connectives to (differentiable) real-valued operators. As in Boolean logic (Kraft & Buell, 1983), the syntax of a t-norm fuzzy rule includes atomic formulas consisting of propositional variables and Logical connectives ($\vee$, $\wedge$) that represents complex compound formulas (Riegel et al., 2020). There is a body of works that employ logic rules for explainable representation learning (Barbiero et al., 2023; Wang et al., 2021; 2023c), reinforcement learning (Jiang & Luo, 2019; Crouse et al., 2021), etc. Our work on TabLog draws inspiration from such work to develop a novel and effective method for TTA of an ensemble of rules for tabular data classification.

3. Preliminaries

We briefly summarize below the test-time training (TTT) paradigm introduced by Liang et al. (2020) which we will adapt to the tabular data setting. Let $\mathcal{D}_S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_S}$ denote a labeled tabular data set of N_S samples from a

source domain S. Each sample $\mathbf{x}_i$ is an M-tuple of feature values $[x_i^1, x_i^2 \ldots x_i^M]$ where $x_i^k \in V^k$, the set of possible values of the k-th feature. In general, V^k can be $\mathbb{R}$ (or a sub-interval of $\mathbb{R}$, or a set of discrete values (in the case of categorical features). y_i denotes the label of instance $\mathbf{x}_i$.

During training, the source model is trained on $\mathcal{D}_S$ to minimize a convex combination of two loss functions:

$$\min_{\boldsymbol{\theta}} l_{ce}(\mathcal{D}_S; \boldsymbol{\theta}) + \lambda l_s(\mathcal{D}_S; \boldsymbol{\theta}), \tag{1}$$

where l_{ce} is the supervised cross-entropy loss and l_s is the self-supervised loss and λ a weight that specifies the relative importance of the two.

At test time, TTT adapts source domain model parameters $\boldsymbol{\theta}$ to obtain target domain parameters $\boldsymbol{\theta}'$ using unlabeled target domain data $\mathcal{D}_T = \{(\mathbf{x}_i)\}_{i=1}^{N_T}$. It should be noted that the feature spaces for the source and target domains are identical; however their distributions are not. TTT entails solving the following optimization problem:

$$\min_{\boldsymbol{\theta}} l_{en}(\mathcal{D}_T; \boldsymbol{\theta}) + \lambda l_s(\mathcal{D}_T; \boldsymbol{\theta}). \tag{2}$$

l_{en} is an entropy loss as defined by Wang et al. (2020).

4. TabLog Algorithm

Key modules of TabLog are shown in Fig. 1. Now we proceed to describe the key components of TabLog: (1) A rule learning model that learns both the atoms and literals used to construct the rules that make up a rule-based classifier for tabular data; (2) A test-time adaptation strategy that adapts the rule-based classifier learned from the source domain data to the target domain. It preserves the logical structure of the rules learned in the source domain, including the atoms or literals as well as the logical connectives that specify their interactions. (3) A self-supervised contrastive loss that enhances the performance of the rule-based classifier in the target domain.

4.1. Rules for Tabular Data Classification

TabLog learns an ensemble of rules as a classifier for tabular data. While it differs in terms of the specific mechanism used for learning the classifier, the form of the classifier and the logical structure of the rules is similar to that introduced in (Friedman & Popescu, 2008). An example of a rule learned by TabLog is given below.

Example 1 Consider the problem of predicting mortality from clinical data. A possible rule could be: (age $\geq$ 80) $\wedge$ (systolic $\leq$ 120) $\wedge$ (diastolic $\leq$ 80) $\wedge$ (gender = female) $\Rightarrow$ Survival. In this example, atoms that evaluate to True or False include (age $\geq$ 80), (systolic $\leq$ 120), etc.

An ensemble of rules classifier consists of a collection or ensemble of such rules, each with an associated weight. Each rule votes for a class label on a given data sample. The sample is assigned the label based on a weighted aggregation of their votes.

Recall that TabLog assumes that the logical structure of the rules remains invariant despite distribution shift from the source domain to the target domain, and that the adaptation to distribution shift can be achieved by re-weighting each rule and adapting parameters e.g., threshold that defines the atoms. Thus, hypothetically, a target domain adapted version of the rule in the above example could be:

Example 2 A possible target domain adaptation of the rule shown in Example 1 could be: (age $\geq$ 84) $\wedge$ (systolic $\leq$ 115) $\wedge$ (diastolic $\leq$ 82) $\wedge$ (gender = female) $\Rightarrow$ Survival.

4.2. Rule Learning in TabLog

TabLog learns an ensemble of rules as follows: First, numerical features are transformed into literals using atom generation module are concatenated with the one-hot encoded categorical features. Then, interactions among atoms are learned using the connective learning module. The predictions of the individual rules are aggregated by the rule voting module to predict the label based on a weighted aggregation of their votes. We next proceed to describe each of these steps in detail.

Encoding Features into Atoms. Boolean and categorical features are simply mapped to their one-hot encoding to obtain the corresponding atoms that are then used to construct the rules. In the case of numeric features, e.g., age, we learn a discretization to partition the feature values into a finite number of bins. Specifically, the range of values of the k-th numerical feature are first split into a disjoint set of $T^k + 1$ bins based on a sequence of increasing thresholds: $\{b_1^k, b_2^k, ..., b_{T^k}^k\}$. This split enables us to map the original numeric values into discrete indices defined by the corresponding thresholds. Each choice of the threshold yields an atom e.g., (age $\geq$ 80) in Example 1, that can be used in rules as we shall see below.

For computational efficiency, we transform the problem of learning thresholds associated with the atoms, into its continuous (differentiable) counterpart. Specifically, we define an indicator function ϕ_l for $x_i^k \geq b_t^k$ as:

$$\phi_l(x_i^k) = \frac{1}{1 + e^{x_i^k - b_t^k}}. \tag{3}$$

Similarly, we define the indicator function ϕ_r for $x_i^k \leq b_t^k$ as:

$$\phi_r(x_i^k) = \frac{1}{1 + e^{-(x_i^k - b_t^k)}}. \tag{4}$$

As we shall see below, these two differentiable functions

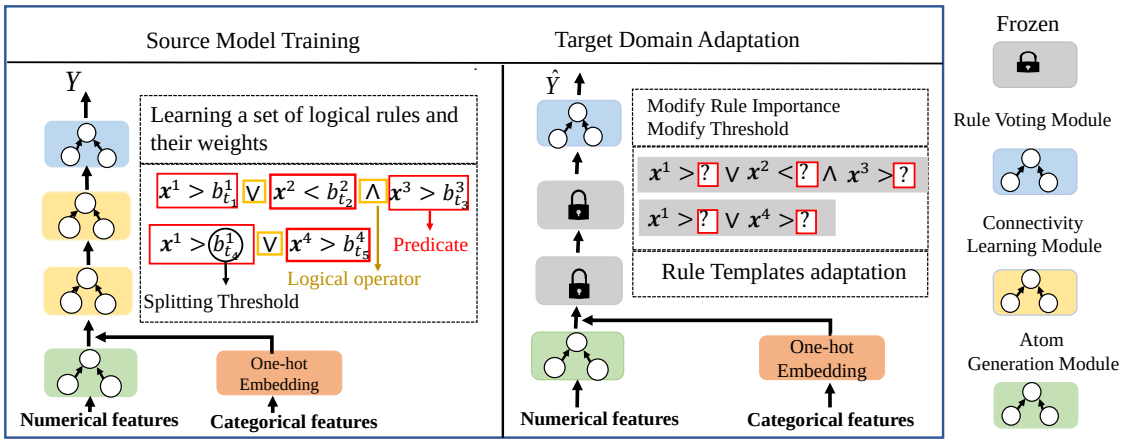

Figure 1. Adaptation for tabular data with logic rules. **Left**: Learning a rule-based classifier in the source domain. First, numerical features are transformed into literals using atom generation module are concatenated with the one-hot encoded categorical features. Then, interactions among atoms will be learned using the connective learning module. The predictions of the individual rules are aggregated by the rule voting module to obtain the prediction. **Right**: Test time adaptation. The learned logical structure of the rules transferred directly from the source domain to the target domain. Only the classifier parameters, namely, the thresholds used for atom generation and rule weights used in rule voting are optimized during TTT.

allow us to optimize the thresholds for all of the numerical variables together with the learning of the rule ensemble. In our implementation of TabLog, the number of bins is a hyperparameter and the initial value of thresholds is sampled from a Gaussian distribution. The atoms resulting from the numeric features, together with the atoms obtained from the one-hot encoding of boolean or categorical features form a complete set of atoms or the primitives used to construct the rules (see below).

Learning Logical Connectives

We leverage logical connectives to model the complicated interactions among atoms. The encoding of features using atoms that evaluate to True or False enables the learning of logical rules in the form of Antecedent $\rightarrow$ Consequent. The Antecedent is a logical expression formed by applying operators such as conjunction ($\wedge$) and disjunction disjunction ($\vee$) to subsets of atoms, while the Consequent corresponds to a class label. However, learning such rules that optimize a desired objective, e.g., classification accuracy, presents a computationally intractable discrete optimization problem (Silva et al., 2020; Chen et al., 2015; Yang et al., 2017). To address this challenge, we leverage differentiable approximate encodings of logical expressions using neural networks (Riegel et al., 2020; Hu et al., 2016; Yang et al., 2022).

Specifically, we adopt the logical neural networks introduced by Riegel et al. (2020) to learn logical rules by optimizing a differentiable function of the weights of a neural network with neurons that approximate the logical $\wedge$ and $\vee$ functions. More precisely, given a set of A atoms $\{a_1, a_2, ..., a_A\}$, their conjunction can be approximated as follows (Riegel et al., 2020):

$$\text{Conj}(\{\alpha_j\}_{j=1}^{A}, \{a_j\}_{j=1}^{A}) = g(1 - \sum_{j=1}^{A} \alpha_j (1 - a_j)), \quad (5)$$

where $g(z) = \max\{0, \min\{z, 1\}\}$ clamps the true values into [0,1], and α_j is the learnable parameter that models the role of a_j in the conjunction (Thus, $\alpha_j = 0$ implies the truth value of a_j has no influence on the evaluation of the conjunctive expression). In a similar fashion, the disjunction of atoms $\{a_1, a_2, ..., a_A\}$ can be approximated by:

$$\text{Disj}(\{\beta_j\}_{j=1}^{A}, \{a_j\}_{j=1}^{A}) = 1 - g(1 - \sum_{j=1}^{A} \beta_j a_j), \quad (6)$$

where β_j is also a learnable parameter that models the role of a_j in the disjunction.

In our logical neural network, each logical connectivity layer consists of multiple Conj and Disj modules. The detailed configuration is provided in Section 5.3. Learned rules can be extracted by analyzing the logical neural network. We add a voting layer on top of the logical neural network to obtain a rule ensemble.

Prediction Using a Rule Ensemble. Recall that the logical neural network implements an ensemble of rules learned from the data. Given C distinct and mutually disjoint class labels, we model the predicted probabilities of the different classes $\hat{\mathbf{p}}^c \in \mathbb{R}^C$ by:

$$\hat{\mathbf{p}}^c = \frac{exp(\mathbf{v}^c \cdot \mathbf{r})}{\sum_c exp(\mathbf{v}^c \cdot \mathbf{r})} \quad (7)$$

where $\mathbf{r} \in \mathbb{R}^K$ is a vector of the K outputs of the penultimate layer of the logical neural network and $\mathbf{v}^c \in \mathbb{R}^K$ denote the learnable weights that associate the them with the c-th class. Thus, there are altogether $C \times K$ learnable weights. The predicted probabilities $\hat{\mathbf{p}}^c$ where $c \in \{1, \cdots C\}$ are used to compute cross-entropy loss in Eq. 1.

4.3. Rule Ensemble Adaptation in TagLog

Contrastive Loss. Self-supervised contrastive learning with carefully chosen pretext task has been shown to be effective for TTT approach to domain adaptation (Liang et al., 2020; 2023; Chen et al., 2022). One such pretext task is classification under feature perturbation or feature corruption which has been shown to be effective for producing classifiers that are robust to distribution shift (Li et al., 2021; Bahri et al., 2021; Yoon et al., 2020). Drawing inspiration from the success of this approach, we introduce a new contrastive loss function that helps the classifier adapt to covariate shift in the target domain relative to the source domain. The proposed loss function is based on the heuristic that the data representation and the learned rules should be robust enough to accommodate minor variations in the input. Specifically, inspired by (Gorishniy et al., 2022), we discretize the features into equal-sized bins based on sample quantiles. For each input sample, we perturb a subset of its features by replacing the value of each perturbed feature by a random value from the same bin as that to which its original value belongs. The resulting pair of samples form a 'positive' pair for contrastive learning. Negative pairs are obtained using a similar process, except that the value of each perturbed feature is replaced by a random value from a bin that is different from that to which its original value belongs. Similar to (Bahri et al., 2021; Wang et al., 2023b), we optimize the InfoNCE contrastive loss. This aims to ensure that the output $\mathbf{r}_i$ and $\hat{\mathbf{r}}_i$ of penultimate layer of the logical neural network for each data sample $\mathbf{x}_i$ and its feature perturbed version $\hat{\mathbf{x}}_i$ to be "close":

$$l_s = -\mathbb{E}_{i \in |\mathcal{D}|} \log \frac{\exp(sim(\mathbf{r}_i, \hat{\mathbf{r}}_i)/\tau)}{\sum_{j \in |\mathcal{D}|} \mathbf{1}_{j \neq i} \exp(sim(\mathbf{r}_i, \mathbf{r}_j)/\tau)}, \quad (8)$$

where τ is the temperature parameter, $\mathbf{1}$ is the indicator function, and sim denotes the cosine similarity.

Test-time Adaptation of Atom Thresholds and Rule Weights. Recall that TabLog uses a source model $f_{\boldsymbol{\theta}}$ learned from labeled source domain tabular data $\mathcal{D}_S$ and aims to adapt the the source model to the target domain in the presence of distribution shift from the source to the target domain. The source and target models in TabLog take the form of weighted rule ensembles implicitly encoded by logical neural network. TabLog assumes that the **log**ical structure of the rules in the source domain are invariant to distribution shift but the numeric parameters associated with the rules, e.g., thresholds that map the numeric features to logical atoms and the rule weights have to change to accommodate distribution shift.

Specifically, we perform test-time adaptation of the model learned from the source domain by minimizing the contrastive loss (see Eq. 2) with respect to the bin threshold parameters $\{b_1^k, b_2^k, ..., b_{T^k}^k\}$ for each of the numeric features and rule weights $\mathbf{v}_1 \cdots \mathbf{v}_C$ of the rule ensemble in Eq.7. The maintaining of model parameters are fixed. This optimization proceeds in a manner identical to source model training procedure except for the fact that the objective function used for training the source model, i.e., that given by Eq. 1 is now replaced by test loss given by Eq. 2 with l_s set to the contrastive loss given by Eq. 8.

5. Experiments

We proceed to describe the experiments of our experiments that demonstrate the effectiveness of the TabLog solution to test-time adaptation of classifiers under distribution shift in tabular data.

5.1. Description of Data Sets

Our experiments used (i) four publicly available benchmark data sets that exhibit natural distribution shifts: ASSISTments, Sepsis, Hospital Readmission, and PhysioNet, available as part of the TableShift benchmark (Gardner et al., 2023); and (ii) four tabular data sets Airbnb, Channel, Jigsaw, and Wine (Shi et al., 2021) subjected to simulated distribution shifts induced by Gaussian noise, uniform noise, and randomly perturbed feature values(Wang et al., 2020). Let j-th feature $\mathbf{x}^j$, where where μ^j and σ^j denote the mean and the standard deviation of the empirical marginal distribution of $\mathbf{x}^j$ estimated from the training set. Then Gaussian noise induced distribution shift is simulated by adding to each feature $\mathbf{x}^j$, Gaussian noise ϵ with $\mathbf{x}^j \leftarrow \mathbf{x}^j + \epsilon \cdot \sigma^j$, where $\epsilon \sim \mathcal{N}(0, 0.1)$; Distribution shift induced by uniform noise is simulated by adding to each feature $\mathbf{x}^j$, uniform noiseϵ with $\mathbf{x}^j \leftarrow \mathbf{x}^j + \epsilon \cdot \sigma^j$. where $\epsilon \sim \mathcal{U}(-0.1, 0.1)$; and distribution shift induced by randomly perturbed feature values is simulated by masking the column corresponding to $\mathbf{x}^j$, and replacing it with a randomly sampled value using a random mask $\mathbf{m}^j$ where the random sampling rate to 0.1.

5.2. Baselines and Evaluation Metrics

Our experiments compare Tablog with following baselines: (1) Classical methods for predictive modeling from tabular data, including Logistic Regression (LR) which is a discriminative counterpart of Naive Bayes, XGBoost (Chen et al., 2015) which builds an ensemble of decision trees, and Catboost which performs boosting with categorical features. (2) Target domain agnostic methods that learn feature representation on source domain that generalizes directly

Table 1. The average accuracy (%), Macro-F1 (%) and their corresponding standard errors for both supervised models and TTA baselines are reported on four TableShift benchmark data sets, which exhibit natural distribution shifts. The results are reported over five different random seeds. Bold denotes the best.

	Method	ASSISTments		Sepsis		Hospital_Readmission		PhysioNet	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Supervised	LR	$43.65_{\pm 0.71}$	$30.38_{\pm 0.32}$	$58.48_{\pm 0.13}$	$48.04_{\pm 0.19}$	$50.61_{\pm 0.10}$	$33.61_{\pm 0.24}$	$87.87_{\pm 0.23}$	$46.05_{\pm 0.10}$
	XGboost	$58.18_{\pm 0.71}$	$54.37_{\pm 0.53}$	$92.46_{\pm 1.08}$	$48.05_{\pm 0.94}$	$62.05_{\pm 0.20}$	$61.93_{\pm 0.15}$	$86.17_{\pm 0.39}$	$45.78_{\pm 0.21}$
	Catboost	$51.46_{\pm 0.35}$	$48.04_{\pm 0.49}$	$92.46_{\pm 0.84}$	$48.04_{\pm 0.55}$	$61.74_{\pm 1.37}$	$61.50_{\pm 1.33}$	$84.90_{\pm 0.29}$	$44.36_{\pm 0.35}$
	VIME	$44.16_{\pm 1.51}$	$41.75_{\pm 1.69}$	$46.31_{\pm 0.33}$	$38.95_{\pm 0.24}$	$51.07_{\pm 0.49}$	$35.19_{\pm 0.30}$	$86.59_{\pm 0.92}$	$46.58_{\pm 1.40}$
	TransTab	$45.01_{\pm 0.41}$	$43.04_{\pm 0.27}$	$45.72_{\pm 0.09}$	$38.91_{\pm 0.11}$	$51.12_{\pm 0.11}$	$35.31_{\pm 0.08}$	$86.03_{\pm 1.57}$	$46.25_{\pm 1.32}$
	Scarf	$43.65_{\pm 0.27}$	$39.38_{\pm 0.31}$	$44.10_{\pm 0.13}$	$38.87_{\pm 0.12}$	$50.62_{\pm 0.51}$	$33.65_{\pm 0.57}$	$85.36_{\pm 1.55}$	$45.95_{\pm 1.36}$
MLP	TENT	$50.60_{\pm 1.27}$	$47.24_{\pm 1.03}$	$48.21_{\pm 0.85}$	$39.10_{\pm 0.61}$	$58.85_{\pm 0.40}$	$56.60_{\pm 0.29}$	$86.00_{\pm 0.94}$	$45.33_{\pm 0.71}$
	EATA	$51.50_{\pm 0.33}$	$48.51_{\pm 0.21}$	$48.21_{\pm 0.54}$	$39.09_{\pm 0.40}$	$59.37_{\pm 0.28}$	$57.01_{\pm 0.32}$	$86.34_{\pm 0.11}$	$45.58_{\pm 0.09}$
	SHOT	$43.65_{\pm 0.24}$	$30.38_{\pm 0.50}$	$51.46_{\pm 0.55}$	$20.40_{\pm 0.72}$	$40.19_{\pm 1.47}$	$29.59_{\pm 1.21}$	$72.17_{\pm 1.54}$	$31.68_{\pm 1.23}$
	TAST	$43.72_{\pm 1.31}$	$30.41_{\pm 0.98}$	$92.46_{\pm 1.10}$	$48.04_{\pm 1.33}$	$50.61_{\pm 2.43}$	$33.60_{\pm 2.07}$	$84.19_{\pm 0.59}$	$46.74_{\pm 0.60}$
FT-trans	TENT	$58.72_{\pm 0.11}$	$55.50_{\pm 0.04}$	$55.14_{\pm 0.94}$	$42.90_{\pm 0.58}$	$57.56_{\pm 0.41}$	$57.33_{\pm 0.30}$	$87.44_{\pm 0.55}$	$45.13_{\pm 0.72}$
	EATA	$58.44_{\pm 0.90}$	$55.16_{\pm 0.85}$	$55.15_{\pm 1.01}$	$43.04_{\pm 0.94}$	$58.44_{\pm 0.04}$	$55.16_{\pm 0.07}$	$87.51_{\pm 0.22}$	$45.22_{\pm 0.16}$
	SHOT	$43.38_{\pm 0.43}$	$42.82_{\pm 0.27}$	$52.26_{\pm 1.79}$	$16.48_{\pm 2.01}$	$59.86_{\pm 0.29}$	$59.60_{\pm 0.11}$	$87.96_{\pm 0.51}$	$46.21_{\pm 0.40}$
	TAST	$61.01_{\pm 0.31}$	$58.44_{\pm 0.25}$	$92.46_{\pm 0.19}$	$48.04_{\pm 0.22}$	$61.17_{\pm 0.44}$	$61.73_{\pm 0.28}$	$88.36_{\pm 0.79}$	$46.92_{\pm 0.52}$
	TabLog	$\mathbf{62.64}_{\pm 0.54}$	$\mathbf{60.96}_{\pm 0.41}$	$\mathbf{98.78}_{\pm 0.19}$	$\mathbf{49.70}_{\pm 0.35}$	$\mathbf{62.92}_{\pm 0.58}$	$\mathbf{62.81}_{\pm 0.69}$	$\mathbf{89.54}_{\pm 0.47}$	$\mathbf{48.03}_{\pm 0.36}$

to the target domain, including VIME (Yoon et al., 2020), which trains a model on pretext tasks of estimating mask vectors from corrupted tabular data and of data reconstruction for learning data representations that are resistant to domain shift, TransTab (Wang & Sun, 2022), a versatile transformer based approach which maps tabular data into an embedding that is robust to distribution shift, and Scarf (Bahri et al., 2021), a contrastive pretraining approach that for maximizing the similarity between a sample and one or more corrupted versions of it obtained by replacing a random subset of its features and replacing them by random draws from the empirical marginal distributions of the respective features. (3) Target domain aware test-time adaptation methods, that fine-tune the parameters of the model trained on the source domain using unlabeled data from the target domain, including Tent (Wang et al., 2020) which optimizes model for confidence as measured by the entropy of its predictions on test data, **EATA** (Niu et al., 2022) which intelligently selects a subset of test samples to minimize entropy loss for test-time adaptation, SHOT (Liang et al., 2020), which learns to extract target domain features that align with the the source domain model, and **TAST** (Jang et al., 2022) which uses nearest neighbors to extract information needed to classify test samples.

We report results of experiments that compare TabLog and the baselines summarized above, several variants of the test-time adaptation methods with different model architectures and different choices of unsupervised pretext tasks and corresponding learning losses. We use accuracy and marco-F1 as our evaluation metrics.

5.3. Implementation Details

Our code uses existing open-source implementations of existing methods. The parameter settings of the baseline models are set based on the respective publications. Since TENT, EATA, SHOT, TASA (Jang et al., 2022) are designed for vision tasks, we experimented with their backbone model replaced with MLP and FT-transformer (Gorishniy et al., 2021). Our experiments were conducted on a Linux server equipped with an A100 GPU. For model optimization, we used SGD as the update rule, with a momentum of 0.9. Unless otherwise specified, the default batch size used was 64. The number of logical neural network layers for learning logical connectives was set to 3 based on exploratory runs that tried values from 1 to 6. The number of conjunction and disjunction modules in each rule learning layer was set to 16. Temperature parameter in Equation 9 was set as 0.1. We report results by averaging over 5 different runs with different random seeds along with standard deviation.

5.4. Experimental Results

Result on Real-World Data With Natural Distribution Shifts. Analyses of the results of our experiments on real-world benchmark data that exhibit distribution shift shown in Table 1 show that:

1. **Across all benchmark data sets, TabLog consistently outperforms the baselines**, lending support to our hypothesis that the logical structure of the rules that make up a rule ensemble remain invariant to distribution shift.

2. **Target domain aware test-time adaptation methods outperform Target domain agnostic methods**, as demonstrated for example, by the consistently superior performance of EATA over Scarf. We hypothesize that the performance gap between these methods is related to the feasibility of learning a feature encoding that both provides sufficient information for accurate classification and is in-

Table 2. The average accuracy (%), Macro-F1 (%) and their corresponding standard errors for both supervised models and TTA baselines are reported on four Tabular data sets. The results reflect average performance over the original data sets and their 3 simulated distribution shifted versions.

	Method	Airbnb		Channel		Jigsaw		Wine	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Supervised	LR	$19.32_{\pm 0.18}$	$10.14_{\pm 0.05}$	$31.05_{\pm 0.69}$	$20.78_{\pm 0.30}$	$87.32_{\pm 1.34}$	$63.73_{\pm 1.41}$	$12.50_{\pm 0.10}$	$1.47_{\pm 0.00}$
	XGboost	$30.74_{\pm 0.05}$	$23.67_{\pm 0.08}$	$30.38_{\pm 0.72}$	$25.32_{\pm 0.57}$	$41.64_{\pm 16.53}$	$39.47_{\pm 14.64}$	$\mathbf{16.92}_{\pm 0.35}$	$\mathbf{4.93}_{\pm 0.11}$
	Catboost	$31.34_{\pm 0.13}$	$21.99_{\pm 0.06}$	$35.48_{\pm 0.45}$	$28.34_{\pm 0.04}$	$41.77_{\pm 16.46}$	$39.61_{\pm 14.58}$	$16.68_{\pm 0.01}$	$4.36_{\pm 0.00}$
	VIME	$15.72_{\pm 0.03}$	$8.04_{\pm 0.00}$	$27.53_{\pm 0.44}$	$18.05_{\pm 0.29}$	$88.17_{\pm 0.24}$	$64.02_{\pm 0.30}$	$14.12_{\pm 0.15}$	$2.64_{\pm 0.09}$
	TransTab	$16.47_{\pm 0.59}$	$8.53_{\pm 0.26}$	$29.64_{\pm 0.84}$	$20.13_{\pm 1.01}$	$87.56_{\pm 1.82}$	$63.02_{\pm 1.55}$	$14.27_{\pm 0.33}$	$2.70_{\pm 0.04}$
	Scarf	$17.20_{\pm 0.03}$	$04.51_{\pm 0.51}$	$34.10_{\pm 0.13}$	$28.87_{\pm 0.12}$	$92.34_{\pm 0.05}$	$49.08_{\pm 3.31}$	$12.28_{\pm 0.01}$	$2.89_{\pm 0.05}$
MLP	TENT	$21.87_{\pm 1.41}$	$15.70_{\pm 0.61}$	$26.22_{\pm 0.14}$	$17.34_{\pm 0.11}$	$94.26_{\pm 0.00}$	$48.52_{\pm 0.00}$	$15.17_{\pm 0.01}$	$2.79_{\pm 0.00}$
	EATA	$23.27_{\pm 0.64}$	$14.21_{\pm 0.33}$	$27.33_{\pm 0.20}$	$18.89_{\pm 0.19}$	$94.21_{\pm 0.05}$	$48.43_{\pm 0.00}$	$12.53_{\pm 0.04}$	$2.10_{\pm 0.00}$
	SHOT	$11.00_{\pm 0.00}$	$5.10_{\pm 0.01}$	$24.73_{\pm 0.00}$	$16.53_{\pm 0.01}$	$74.20_{\pm 0.18}$	$47.14_{\pm 0.00}$	$10.9_{\pm 0.05}$	$3.15_{\pm 0.01}$
	TAST	$34.57_{\pm 1.08}$	$28.42_{\pm 0.64}$	$35.63_{\pm 0.56}$	$28.89_{\pm 0.97}$	$94.26_{\pm 0.00}$	$48.52_{\pm 0.00}$	$12.03_{\pm 0.02}$	$2.93_{\pm 0.00}$
FT-trans	TENT	$24.16_{\pm 0.00}$	$19.49_{\pm 0.52}$	$26.96_{\pm 0.13}$	$21.27_{\pm 0.14}$	$75.96_{\pm 6.88}$	$46.02_{\pm 0.99}$	$14.16_{\pm 0.11}$	$2.37_{\pm 0.08}$
	EATA	$25.53_{\pm 0.00}$	$20.17_{\pm 0.57}$	$28.00_{\pm 0.01}$	$21.21_{\pm 0.12}$	$76.72_{\pm 4.85}$	$46.94_{\pm 0.64}$	$13.53_{\pm 0.14}$	$2.10_{\pm 0.22}$
	SHOT	$16.36_{\pm 0.11}$	$10.67_{\pm 0.02}$	$26.53_{\pm 0.24}$	$18.54_{\pm 0.14}$	$65.32_{\pm 4.21}$	$43.14_{\pm 0.15}$	$8.81_{\pm 0.20}$	$1.16_{\pm 0.13}$
	TAST	$65.52_{\pm 8.01}$	$63.98_{\pm 8.85}$	$36.37_{\pm 0.60}$	$29.46_{\pm 1.20}$	$\mathbf{95.91}_{\pm 0.00}$	$\mathbf{74.98}_{\pm 1.01}$	$12.49_{\pm 0.12}$	$2.60_{\pm 0.09}$
	TabLog	$\mathbf{66.49}_{\pm 0.92}$	$\mathbf{64.52}_{\pm 0.78}$	$\mathbf{37.96}_{\pm 0.51}$	$\mathbf{31.05}_{\pm 0.88}$	$95.17_{\pm 0.82}$	$74.26_{\pm 0.50}$	$15.98_{\pm 0.47}$	$3.42_{\pm 0.36}$

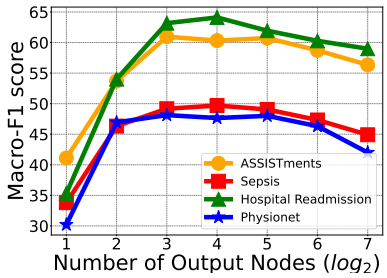

Figure 2. Effect of the Number of Output Nodes in the Penultimate Layer.

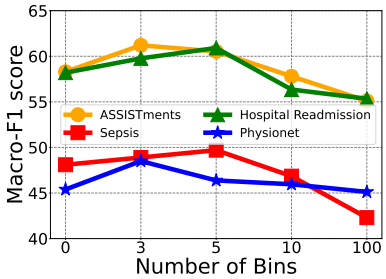

Figure 3. Effect of the Number of Bins.

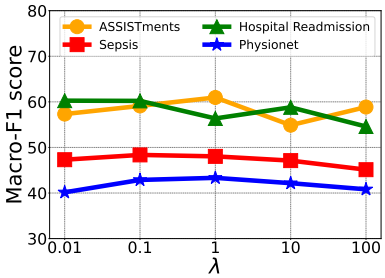

Figure 4. Parameter Sensitivity Analysis.

variant with respect to a broad range of distribution shifts. Adapting the representation or predictive model based on distribution-shifted target data is in general easier than having to anticipate the adaptation needed before seeing the target domain data.

3. **Architectural choice has a significant impact on the effectiveness of different apporoaches to test-time domain adaptation.** For example, TENT, EATA, SHOT and TAST with FT-transformer architecture outperform or are competitive with their multi-layer perceptron (MLP) counterparts. We hypothesize that effective test-time domain adaptation demands greater model complexity or model capacity to accommodate complex distribution shifts that heterogeneous and complex tabular data are likely to exhibit.

4. **Self-Supervised Learning (SSL) on target domain with carefully chosen pretext task appears to improve test-time adaptatin performance.** For example, VIME outperforms Scarf on the ASSISTments, Hospital Readmission, and PhysioNet datasets. TAST outperforms TENT, EATA, and SHOT. Although TENT is effective in calibrating Batch Normalization (BN) statistics for vision data, it substantially underperforms TAST on the ASSISTments data. We hypothesize that the heterogeneous property of tabular data adds more complexity for BN calibration. TabLog outperforms all other methods, including TAST on all benchmarks suggesting the real-world effectiveness of TabLog's assumption of invariance of the logical structure of rules of rule ensemble classifier and its choice of pretext task for test-time domain adaptation. An example of a rule ensemble learned by TagLog is shown in Table 2.

Results on Simulated Distribution Shifted Data. Table 3 shows the the performance of TabLog on Airbnb, Channel, Jigsaw, and Wine data sets (Shi et al., 2021) subjected to simulated distribution shifts induced by Gaussian noise, uniform noise, and randomly missing feature values. For space limitation, we present the mean average results on these three types of simulated distribution shift.

1. **TabLog can effectively adapt to different types of distribution shifts**, as shown by the superior or comparable performance of TabLog relative to all other methods.

Table 3. Examples of learned rules with rule weights on the ASSISMent dataset.

Dataset	Weight	Rules
Source	0.1217	(MasterySectionType_1) $\vee$ (BORED $<$ 0.859) $\vee$ (first_response $<$ 0.326) $\vee$ (TutorMode $<$ 0.132)
	0.0598	(AlgebraType_2) $\wedge$ (BORED $>$ 0.012) $\wedge$ (BORED $<$ 0.859) $\vee$ (CONFUSED $<$ 0.331)
Target	0.0934	(MasterySectionType_1) $\vee$ (BORED $<$ 1.103) $\vee$ (first_response $<$ 0.714) $\vee$ (TutorMode $<$ 0.297)
	0.832	(AlgebraType_2) $\wedge$ (BORED $>$ 0.330) $\wedge$ (BORED $<$ 1.103) $\vee$ (CONFUSED $<$ 1.899)

The only other method that approaches the performance of TabLog is TAST, which combines multiple mechanisms (Jang et al., 2022) and is considerably more complex than TabLog.

2. **All models struggle to perform satisfactorily in the presence of distribution shift on small data sets.** We hypothesize that the challenges of learning from small data sets is further exacerbated by the additional requirements of adapting to distribution shift.

3. **Domain aware test-time adaptation methods outperform Target domain agnostic methods**. This finding is consistent with our observations from experiments on real-world data with natural distribution shift, confirming the advantages of methods that adapt the representation or predictive model based on distribution-shifted target data over methods that have to anticipate the adaptation needed before they see the target domain data.

5.5. Additional Analyses of TabLog

Effect of the Number of Rules in the Rule Ensemble. Figure 2 shows the performance of TabLog as a function of the number of output nodes (on a log to the base 2 scale) in the penultimate layer of the logical neural network that learns the rules in our rule ensemble classifier. Recall that the the number of output nodes is a proxy for the number of rules. We find that the performance of TabLog increases until it plateaus with the number of outputs of the logical neural network set between 8 and 32 and then begins to decrease. This is consistent with what we expect to occur with any predictive model – performance increases as the model complexity increases until over-fitting kicks in leading to a drop in performance.

Effect of the Number of Bins Used to Drive Sampling of Feature Perturbed Samples for Contrastive Learning. Figure 3 shows the performance of TabLog as a function of the number of bins T used to generate 'positive' and 'negative' pairs, of samples used for contrastive learning. In general, we expect the optimal number of bins to be data set dependent. If T is too small, the bins are likely to be less homogeneous in terms of class labels thereby violating the key assumption behind contrastive learning – that similar samples should have similar class labels. On the other hand,

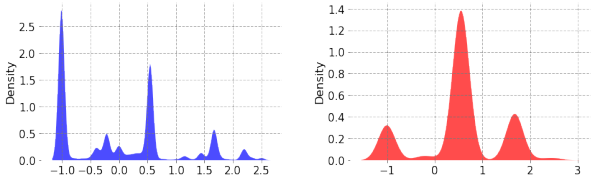

Figure 5. Distribution Visualization of the BORED feature.

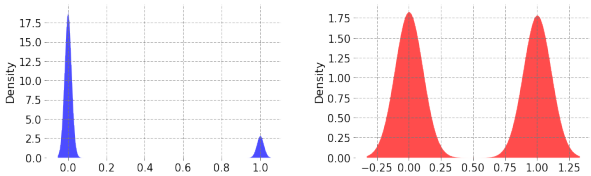

Figure 6. Distribution Visualization of the TutorMode feature.

if T is too large, 'positive' pairs are not sufficiently diverse, thereby negatively impacting the performance of TabLog. In our experiments, 3 o 5 bins appear to yield the best observed performance.

Effect of λ. Figure 4 shows the performance of TabLog as a function of λ in Eg. 2. Smaller the value of λ, the lower the relative importance of the self-supervised loss in Eq. 1 and 2. While in general, the optimal choice of λ is likely to be data dependent, we find that $\lambda \approx 0.1$ appears to yield close to optimal performance which remains relatively constant in the neighborhood of $\lambda \approx 0.1$ on all of the real-world data sets with natural distribution shift.

Deeper Dive into Rule Adaptation. Table 3 shows a subset of the learned rules along with their weights on source domain and target domain from ASSISMent data set. Figures 5 and 6 show the source and target domain distributions (shown in red and blue colors respectively) of two numerical features, namely 'BORED' and 'TutorMode'. Both features exhibit significant distribution shifts. We note that both features exhibit multi-modal distributions in both the source and the target domains. It is also clear that bin boundaries learned from source domain data determine the atoms or propositions that make up the classification rules must be adapted to accommodate the distribution shift. This further confirms one of the key intuitions that informed the design of TabLog. Examination of two of the source domain rules and their target domain adapted counterparts shown in Table 3 confirms that TabLog can effectively adapt both the bin boundaries (thresholds used to discretize numeric features)

and the weights associated with the rules.

6. Conclusion and Future Works

We considered the problem of test-time adaptation of predictive models trained on tabular data. Effective solution of this problem requires adapting a source tabular model to a shifted and unlabeled target data during the test time, without access to the source data. We introduced TabLog, a novel solution to this problem in the form of an efficient end-to-end pipeline for training and test-time adaptation of a rule ensemble for tabular data classification. TabLog works under the assumption that the logical structure of the rules that make up the rule ensemble remains invariant to distribution shift whereas the atoms or literals used to construct the rules and the weights associated with the rules in the rule ensemble need to change in response to distribution shift. Our analyses of results of extensive experiments on using several benchmark data sets show that TabLog outperforms or is competitive with the state-of-the-art methods for test-time adaptation of predictive models trained on tabular data.

Some promising directions for further research include investigation of test-time adaptation methods that: accommodate alternative approaches to mapping numeric features into logical propositions; handle shifts not only in the distributions of individual features, but also their non-linear combinations; cope with label shifts; adapt to the emergence of new, previously unobserved features; accommodate continual domain adaptation with tabular data; extend TabLog or similar methods to tabular time series or tabular longitudinal data.

Impact Statement

This paper presents work that is primarily focused on machine learning advances domain adaptation for tabular data. Applications of the resulting methods, like any other applications of machine learning, may have societal impacts. However, such impacts are likely to be very much application-dependent. Because the primary focus of the work presented here is algorithmic, we do not find the need to speculate about them here.

Acknowledgements

We thank the anonymous reviewers for their critical reviews and suggestions that have helped us improve the paper. This work was funded in part by grants from the National Science Foundation (2226025, 2041759) to Vasant Honavar, and from the National Center for Advancing Translational Sciences, and the National Institutes of Health (UL1 TR002014) to the Pennsylvania State University.

References

Arik, S. Ö. and Pfister, T. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Bahri, D., Jiang, H., Tay, Y., and Metzler, D. Scarf: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2021.

Barbiero, P., Ciravegna, G., Giannini, F., Zarlenga, M. E., Magister, L. C., Tonda, A., Lio, P., Precioso, F., Jamnik, M., and Marra, G. Interpretable neural-symbolic concept reasoning. 2023.

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Chen, C., Chen, Z., Jiang, B., and Jin, X. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3296–3303, 2019.

Chen, D., Wang, D., Darrell, T., and Ebrahimi, S. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Chen, T. and Guestrin, C. 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.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

Cignoli, R., Esteva, F., Godo, L., and Torrens, A. Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft computing*, 4:106–112, 2000.

Crouse, M., Abdelaziz, I., Makni, B., Whitehead, S., Cornelio, C., Kapanipathi, P., Srinivas, K., Thost, V., Witbrock, M., and Fokoue, A. A deep reinforcement learning approach to first-order logic theorem proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6279–6287, 2021.

De Geus, A. J. and Cohen, W. A rule-based system for optimizing combinational logic. *IEEE Design & Test of Computers*, 2(4):22–32, 1985.

Fang, Y., Yap, P.-T., Lin, W., Zhu, H., and Liu, M. Source-free unsupervised domain adaptation: A survey. *Neural Networks*, pp. 106230, 2024.

Friedman, J. H. and Popescu, B. E. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3): 916–954, 2008.

Gandelsman, Y., Sun, Y., Chen, X., and Efros, A. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

Gardner, J., Popovic, Z., and Schmidt, L. Benchmarking distribution shift in tabular data with tableshift. *arXiv preprint arXiv:2312.07577*, 2023.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., and Lee, S.-J. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021.

Gorishniy, Y., Rubachev, I., and Babenko, A. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35: 24991–25004, 2022.

Goyal, S., Sun, M., Raghunathan, A., and Kolter, J. Z. Test time adaptation via conjugate pseudo-labels. *Advances in Neural Information Processing Systems*, 35:6204–6218, 2022.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2410–2420, 2016.

Jang, M., Chung, S.-Y., and Chung, H. W. Test-time adaptation via self-training with nearest neighbor information. In *The Eleventh International Conference on Learning Representations*, 2022.

Jiang, Z. and Luo, S. Neural logic reinforcement learning. In *International conference on machine learning*, pp. 3110–3119. PMLR, 2019.

Kraft, D. H. and Buell, D. A. Fuzzy sets and generalized boolean retrieval systems. *International journal of man-machine studies*, 19(1):45–56, 1983.

Kundu, J. N., Venkat, N., Babu, R. V., et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4544–4553, 2020.

Li, P., Li, D., Li, W., Gong, S., Fu, Y., and Hospedales, T. M. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8886–8895, 2021.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pp. 6028–6039. PMLR, 2020.

Liang, J., Hu, D., Wang, Y., He, R., and Feng, J. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8602–8617, 2021.

Liang, J., He, R., and Tan, T. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023.

Nilsson, N. J. Probabilistic logic. *Artificial intelligence*, 28 (1):71–87, 1986.

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. Mit Press, 2022.

Ren, W., Wang, P., Li, X., Hughes, C. E., and Fu, Y. Semi-supervised drifted stream learning with short lookback. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1504–1513, 2022.

Ren, W., Zhao, T., Qin, W., and Liu, K. T-sas: Toward shift-aware dynamic adaptation for streaming data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4244–4248, 2023.

Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I. Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., et al. Logical neural networks. *arXiv preprint arXiv:2006.13155*, 2020.

Shi, X., Mueller, J., Erickson, N., Li, M., and Smola, A. J. Benchmarking multimodal automl for tabular data with text fields. *arXiv preprint arXiv:2111.02705*, 2021.

Silva, A., Gombolay, M., Killian, T., Jimenez, I., and Son, S.-H. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *International conference on artificial intelligence and statistics*, pp. 1855–1865. PMLR, 2020.

Somepalli, G., Schwarzschild, A., Goldblum, M., Bruss, C. B., and Goldstein, T. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS 2022 First Table Representation Workshop*, 2022.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Ucar, T., Hajiramezanali, E., and Edwards, L. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020.

Wang, J. and Chen, Y. Transfer learning for computer vision. In *Introduction to Transfer Learning: Algorithms and Practice*, pp. 265–273. Springer, 2022.

Wang, S., Zhang, D., Yan, Z., Zhang, J., and Li, R. Feature alignment and uniformity for test time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20050–20060, 2023a.

Wang, Y., Wang, Z., Lin, Y., Guo, J., Halim, S., and Khan, L. Dual contrastive learning framework for incremental text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 194–206, 2023b.

Wang, Z. and Sun, J. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

Wang, Z., Zhang, W., Liu, N., and Wang, J. Scalable rule-based representation learning for interpretable classification. *Advances in Neural Information Processing Systems*, 34:30479–30491, 2021.

Wang, Z., Zhang, W., Liu, N., and Wang, J. Learning interpretable rules for scalable data representation and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023c.

Yang, F., Yang, Z., and Cohen, W. W. Differentiable learning of logical rules for knowledge base reasoning. *Advances in Neural Information Processing Systems*, 30, 2017.

Yang, Z., Lee, J., and Park, C. Injecting logical constraints into neural networks via straight-through estimators. In *International Conference on Machine Learning*, pp. 25096–25122. PMLR, 2022.

Yoon, J., Zhang, Y., Jordon, J., and van der Schaar, M. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zhang, W., Liu, Y., Wang, Z., and Wang, J. Learning to binarize continuous features for neuro-rule networks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4584–4592, 2023.