Deep Neural Networks and Tabular Data: A Survey

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci

***Abstract*— Heterogeneous tabular data are the most commonly used form of data and are essential for numerous critical and computationally demanding applications. On homogeneous datasets, deep neural networks have repeatedly shown excellent performance and have therefore been widely adopted. However, their adaptation to tabular data for inference or data generation tasks remains highly challenging. To facilitate further progress in the field, this work provides an overview of state-of-the-art deep learning methods for tabular data. We categorize these methods into three groups: data transformations, specialized architectures, and regularization models. For each of these groups, our work offers a comprehensive overview of the main approaches. Moreover, we discuss deep learning approaches for generating tabular data and also provide an overview over strategies for explaining deep models on tabular data. Thus, our first contribution is to address the main research streams and existing methodologies in the mentioned areas while highlighting relevant challenges and open research questions. Our second contribution is to provide an empirical comparison of traditional machine learning methods with 11 deep learning approaches across five popular real-world tabular datasets of different sizes and with different learning objectives. Our results, which we have made publicly available as competitive benchmarks, indicate that algorithms based on gradient-boosted tree ensembles still mostly outperform deep learning models on supervised learning tasks, suggesting that the research progress on competitive deep learning models for tabular data is stagnating. To the best of our knowledge, this is the first in-depth overview of deep learning approaches for tabular data; as such, this work can serve as a valuable starting point to guide researchers and practitioners interested in deep learning with tabular data.**

***Index Terms*— Benchmark, deep neural networks, discrete data, heterogeneous data, interpretability, probabilistic modeling, survey, tabular data, tabular data generation.**

I. Introduction

EVER-INCREASING computational resources and the availability of large, labeled datasets have accelerated the success of deep neural networks [1], [2]. In particular, architectures based on convolutions, recurrent mechanisms [3], [4], or transformers [5] have led to unprecedented performance in a multitude of domains. Although deep learning methods perform outstandingly well for classification or data generation tasks on homogeneous data (e.g., image, audio, and text data), tabular data still pose a challenge to deep learning models [6], [7], [8]. Tabular data—in contrast to image or language data—are heterogeneous, leading to dense numerical and sparse categorical features. Furthermore, the correlation among the features is weaker than the one introduced through spatial or semantic relationships in image or speech data. Hence, it is necessary to discover and exploit relations without relying on spatial information [9]. Therefore, Kadra et al. [10] called tabular datasets the "last unconquered castle" for deep neural network models.

Heterogeneous data are the most commonly used form of data [8], and it is ubiquitous in many crucial applications, such as medical diagnosis based on patient history [11], [12], [13], predictive analytics for financial applications (e.g., risk analysis, estimation of creditworthiness, the recommendation of investment strategies, and portfolio management) [14], click-through rate (CTR) prediction [15], user recommendation systems [16], [17], customer churn prediction [18], cybersecurity [19], fraud detection [20], psychology [21], anomaly detection [22], [23], [24], and so forth. In all these applications, a boost in predictive performance and robustness may have considerable benefits for both end users and companies that provide such solutions. Simultaneously, this requires handling many data-related pitfalls, such as noise, impreciseness, different attribute types and value ranges, or the missing value problem and privacy issues.

Meanwhile, deep neural networks offer multiple advantages over traditional machine learning methods. First, these methods are highly flexible [25], allow for efficient and iterative training, and are particularly valuable for AutoML [26], [27]. Second, tabular data generation is possible using deep neural networks and can, for instance, help mitigate class imbalance problems [28]. Third, neural networks can be deployed for multimodal learning problems where tabular data can be one of many input modalities [29], [30], for tabular data distillation [31], [32], for federated learning [33], and in many more scenarios.

Successful deployments of data-driven applications require solving several tasks, among which we identified three core challenges: 1) inference; 2) data generation; and 3) interpretability. The most crucial task is inference, which is concerned with making predictions based on past observations. While a powerful predictive model is critical for all the applications mentioned in the previous paragraph, the interplay between tabular data and deep neural networks goes beyond simple inference tasks. Before a predictive model can even be trained, the training data usually need to be preprocessed. This is where data generation plays a crucial role, as one of the standard deployment steps involves the imputation of missing values [34], [35] and the rebalancing of the dataset [36], [37] (i.e., equalizing sample sizes for different classes). Furthermore, it might be simply impossible to use the actual data due to privacy concerns, e.g., in financial or medical

Manuscript received 21 February 2022; revised 29 June 2022, 24 October 2022, and 28 November 2022; accepted 12 December 2022. Date of publication 23 December 2022; date of current version 4 June 2024. *(Corresponding authors: Vadim Borisov; Tobias Leemann.)*

The authors are with the Data Science and Analytics Research (DSAR) Group, University of Tübingen, 72070 Tübingen, Germany (e-mail: vadim.borisov@uni-tuebingen.de; tobias.leemann@uni-tuebingen.de).

Digital Object Identifier 10.1109/TNNLS.2022.3229161

applications [38], [39]. Thus, to tackle the data preprocessing and privacy challenges, probabilistic tabular data generation is essential. Finally, with stricter data protection laws such as California Consumer Privacy Act (CCPA) [40] and the European General Data Protection Regulation (EU GDPR) [41], which both mandate a right to explanations for automated decision systems (e.g., in the form or recourse [42]), interpretability is becoming a key aspect for predictive models used for tabular data [43], [44]. During deployment, interpretability methods also serve as a valuable tool for model debugging and auditing [45].

Evidently, apart from the core challenges of inference, generation, and interpretability, there are several other important subfields, such as working with data streams, distribution shifts, as well as privacy and fairness considerations that should not be neglected. Nevertheless, to navigate the vast body of literature, we focus on the identified core problems and thoroughly review the state of the art in this work. We will briefly discuss the remaining topics at the end of this survey.

Beyond reviewing current literature, we think that an exhaustive comparison between existing deep learning approaches for heterogeneous tabular data is necessary to put reported results into context. The variety of benchmarking datasets and the different setups often prevent the comparison of results across papers. In addition, important aspects of deep learning models, such as training and inference time, model size, and interpretability, are usually not discussed. We aim to bridge this gap by providing a comparison of the surveyed inference approaches with classical—yet very strong—baselines such as XGBoost [46]. We open-source our code, allowing researchers to reproduce and extend our findings.

In summary, the aims of this survey are to provide the following:

1) a thorough review of existing scientific literature on deep learning for tabular data;
2) a taxonomic categorization of the available approaches for classification and regression tasks on heterogeneous tabular data;
3) a presentation of the state of the art and promising paths toward tabular data generation;
4) an overview of existing explanation approaches for deep models for tabular data;
5) an extensive empirical comparison of traditional machine learning methods and deep learning models on multiple real-world heterogeneous tabular datasets;
6) a discussion on the main reasons for the limited success of deep learning on tabular data;
7) a list of open challenges related to deep learning for tabular data.

Accordingly, this survey is structured as follows. We discuss related works in Section II. To introduce the reader to the field, in Section III, we provide definitions of the key terms, a brief outline of the domain's history, and propose a unified taxonomy of current approaches to deep learning with tabular data. Section IV covers the main methods for modeling tabular data using deep neural networks. Section V presents an overview on tabular data generation using deep neural networks. An overview of explanation mechanisms for deep models for tabular data is presented in Section VI. In Section VII, we provide an extensive empirical comparison of machine and deep learning methods on real-world data, which also involves model size, runtime, and interpretability. In Section VIII, we summarize the state of the field and give future perspectives. Finally, we outline several open research questions before concluding in Section IX.

II. Related Work

To the best of our knowledge, there is no study dedicated exclusively to the application of deep neural networks to tabular data, spanning the areas of supervised and unsupervised learning, data synthesis, and interpretability. Prior works cover some of these aspects, but none of them systematically discusses the existing approaches in the broadness of this survey.

However, there are some works that cover parts of the domain. There is a comprehensive analysis of common approaches for categorical data encoding as a preprocessing step for deep neural networks by Hancock and Khoshgoftaar [47]. The authors compared existing methods for categorical data encoding on various tabular datasets and different deep learning architectures. We discuss the key categorical data encoding methods in Section IV-A1.

A recent survey by Sahakyan et al. [43] summarizes explanation techniques in the context of tabular data. Hence, we do not provide a detailed discussion of explainable machine learning for tabular data in this article. However, for the sake of completeness, we present some of the most relevant works in Section VI and highlight open challenges in this area.

Gorishniy et al. [48] empirically evaluated a large number of state-of-the-art deep learning approaches for tabular data on a wide range of datasets. He et al. [49] demonstrated that a tuned deep neural network model with a ResNet-like architecture shows comparable performance to some state-of-the-art deep learning approaches for tabular data.

Recently, Shwartz-Ziv and Armon [8] published a study on several different deep models for tabular data, including TabNet [6], NODE [7], and Net-DNF [50]. In addition, they compared deep learning approaches to gradient boosting decision tree (GBDT) algorithms regarding accuracy, training effort, inference efficiency, and hyperparameter optimization time. They observed that deep models had the best results on their chosen datasets, and however, not one single deep model could outperform all the others in general. The deep models were challenged by GBDTs, leading the authors to conclude that efficient tabular data modeling using deep neural networks is still an open research problem. In the face of this evidence, we aim to integrate the necessary background for future research on the inference problem and on the intertwined challenges of generation and explainability into a single work.

III. Tabular Data and Deep Neural Networks

A. Definitions

In this section, we give definitions for central terms used in this work. We also provide pointers to the original works for more detailed explanations of the methods.

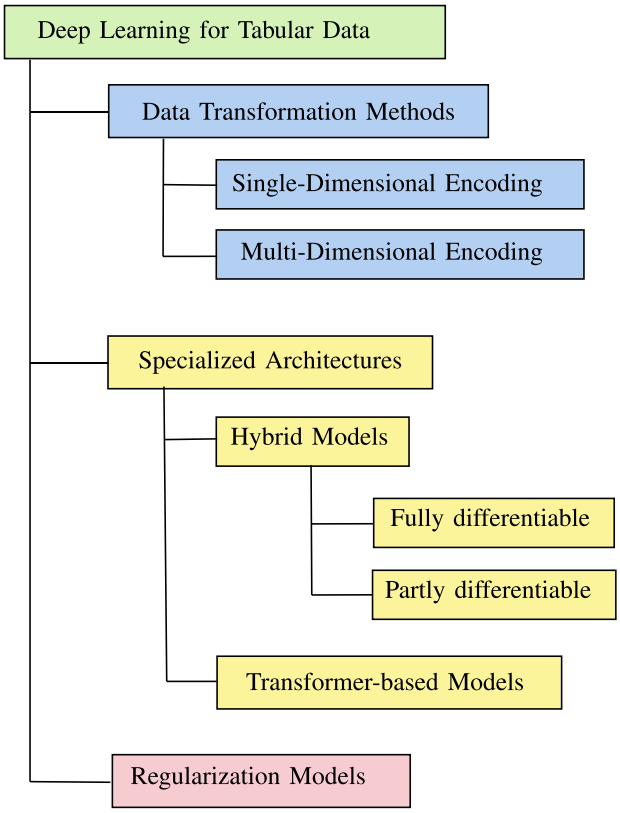

Fig. 1. Unified taxonomy of deep neural network models for heterogeneous tabular data.

The key concept in this survey is a (deep) neural network. Unless stated otherwise we use this concept as a synonym for feedforward networks, as described in [2], and name the concrete model whenever we deviate from this concept. A deep neural network defines mapping $\hat{f}$

$$y = f(x) \approx \hat{f}(x; W) \tag{1}$$

that learns the value of the model parameters W (i.e., the "weights" of a neural network) that results in the best approximation of the true underlying and unknown function f. In this case, x is a multidimensional data sample (i.e., $x \in \mathbb{R}^n$) with corresponding target y (where typically, $y \in \mathbb{R}^k$ for k classes and $y \in \mathbb{R}$ for regression tasks) from a dataset of tuples $\{(x_i, y_i)\}_{i \in \mathcal{I}}$. The network is called feedforward if the input information flows in one direction to the output without any feedback connections.

Throughout this survey, we focus on heterogeneous data that usually contain a variety of attribute types. These include both continuous and discrete attributes of different types (e.g., binary values, ordinal values, and high-cardinality categorical values). This is fundamentally different from homogeneous data modalities, such as images, audio, or text data where only a single feature type is present.

Categorical variables are an attribute type of particular importance. According to Lane's definition [51], categorical variables are qualitative values. They "do not imply a numerical ordering," unlike quantitative values, which are "measured in terms of numbers." Usually, a categorical variable can take one out of a limited set of values. Examples of typical categorical variables include gender, user_id, product_type and topic.

TABLE I

EXAMPLE OF A HETEROGENEOUS TABULAR DATASET. HERE, WE SHOW FIVE SAMPLES WITH SELECTED VARIABLES FROM THE ADULT DATASET [54]. SECTION VII-A PROVIDES FURTHER DETAILS ON THIS DATASET

Age	Education	Occupation	Sex	Income
39	Bachelors	Adm-clerical	Male	$\leq$50K
50	Bachelors	Exec-managerial	Male	$>$50K
38	HS-grad	Handlers-cleaners	Male	$\leq$50K
53	11th	Handlers-cleaners	Male	$\leq$50K
28	Bachelors	Prof-specialty	Female	$>$50K

Tabular data, sometimes also called structured data [52], are the subcategory of the heterogeneous data format that is usually presented in a table [53] with data points as rows and features as columns. In summary, for the scope of this work, we refer to a dataset with a fixed number of features that are either continuous or categorical as tabular. Each data point can be understood as a row in the table, or—taking a probabilistic view—as a sample from the unknown joint distribution. An illustrative example of five rows of heterogeneous, tabular data is provided in Table I.

B. Brief History of Deep Learning on Tabular Data

Tabular data are one of the oldest forms of data to be statistically analyzed. Before digital collection of text, images, and sound was possible, almost all data were tabular [55], [56], [57]. Therefore, it was the target of early machine learning research [58]. However, deep neural networks became popular in the digital age and were further developed with a focus on homogeneous data. In recent years, various supervised, self-supervised, and semisupervised deep learning approaches have been proposed, which explicitly address the issue of tabular data modeling again. Early works mostly focused on data transformation techniques for preprocessing [59], [60], which are still important today [47].

A huge stimulus was the rise of e-commerce, which demanded novel solutions, especially in advertising [15], [61]. These tasks required fast and accurate estimation on heterogeneous datasets with many categorical variables, for which the traditional machine learning approaches are not well suited (e.g., categorical features that have high cardinality can lead to very sparse high-dimensional feature vectors and nonrobust models). As a result, researchers and data scientists started looking for more flexible solutions, e.g., those based on deep neural networks, that can capture complex nonlinear dependencies in the data.

In particular, the CTR prediction problem has received a lot of attention [15], [62]. A large variety of approaches were proposed, most of them relying on specialized neural network architectures for heterogeneous tabular data.

A more recent line of research, sparked by Shavitt and Segal [63], evolved based on the idea that regularization may

improve the performance of deep neural networks on tabular data [10]. This has led to an intensification of research on regularization approaches.

Due to the tremendous success of attention-based approaches such as transformers on textual [64] and visual data [65], [66], researchers have recently also started applying attention-based methods and self-supervised learning techniques to tabular data. After the introduction of transformer architectures to the field of tabular data [6], a lot of research effort has focused on transformer architectures that can be successfully applied to very large tabular datasets.

C. Challenges of Learning With Tabular Data

As we have mentioned in Section II, deep neural networks often perform less favorably compared to more traditional machine learning methods (e.g., tree-based methods) when dealing with tabular data. However, it is often unclear why deep learning cannot achieve the same level of predictive quality as in other domains such as image classification and natural language processing. In the following, we identify and discuss four possible reasons.

1) *Low-Quality Training Data:* Data quality is a common issue with real-world tabular datasets. They often include missing values [34], extreme data (outliers) [67], and erroneous or inconsistent data [68] and have a small overall size relative to the high-dimensional feature vectors generated from the data [69]. Also, due to the expensive nature of data collection, tabular data are frequently class-imbalanced. These challenges affect all machine learning algorithms; however, most of the modern decision tree-based algorithms can handle missing values or different/extreme variable ranges internally by looking for appropriate approximations and split values [46], [70], [71].
2) *Missing or Complex Irregular Spatial Dependencies:* There is often no spatial correlation between the variables in tabular datasets [72] or the dependencies between features are rather complex and irregular. When working with tabular data, the structure and relationships between its features have to be learned from scratch. Thus, the inductive biases used in popular models for homogeneous data, such as convolutional neural networks, are unsuitable for modeling this data type [50], [73], [74].
3) *Dependency on Preprocessing:* A key advantage of deep learning on homogeneous data is that it includes an implicit representation learning step [2], so only a minimal amount of preprocessing or explicit feature construction is required. However, for tabular data and deep neural networks, the performance may strongly depend on the selected preprocessing strategy [75]. Handling the categorical features remains particularly challenging [47] and can easily lead to a very sparse feature matrix (e.g., by using a one-hot encoding scheme) or introduce a synthetic ordering of previously unordered values (e.g., by using an ordinal encoding scheme). Finally, preprocessing methods for deep neural networks may lead to information loss, leading to a reduction in predictive performance [76].
4) *Importance of Single Features:* While typically changing the class of an image requires a coordinated change in many features, i.e., pixels, the smallest possible change of a categorical (or binary) feature can entirely flip a prediction on tabular data [63]. In contrast to deep neural networks, decision-tree algorithms can handle varying feature importance exceptionally well by selecting a single feature and appropriate threshold (i.e., splitting) values and "ignoring" the rest of the data sample. Shavitt and Segal [63] have argued that individual weight regularization may mitigate this challenge and motivated more work in this direction [10].

With these four fundamental challenges in mind, we continue by organizing and discussing the strategies developed to address them. We start by developing a suitable taxonomy.

D. Unified Taxonomy

In this section, we introduce a taxonomy of approaches that allows for a unified view of the field. We divide the works from the deep learning with tabular data literature into three main categories: data transformation methods, specialized architectures, and regularization models. In Fig. 1, we provide an overview of our taxonomy of deep learning methods for tabular data.

1) Data Transformation Methods: The methods in the first group transform categorical and numerical data. This is usually done to enable deep neural network models to better extract the information signal. Methods from this group do not require new architectures or adaptations of the existing data processing pipeline. Nevertheless, the transformation step comes at the cost of an increased preprocessing time. This might be an issue for high-load systems [77], particularly in the presence of categorical variables with high cardinality and growing dataset size. We can further subdivide this area into single-dimensional encodings and multidimensional encodings. The former encodings are employed to transform each feature independently while the latter encoding methods map an entire record to another representation.

2) Specialized Architectures: The biggest share of works investigates specialized architectures and suggests that a different deep neural network architecture is required for tabular data. Two types of architectures are of particular importance: hybrid models fuse classical machine learning approaches (e.g., decision trees) with neural networks, while transformer-based models rely on attention mechanisms.

3) Regularization Models: Finally, the group of regularization models claims that one of the main reasons for the moderate performance of deep learning models on tabular data is their extreme nonlinearity and model complexity. Therefore, strong regularization schemes are proposed as a solution. They are mainly implemented in the form of special-purpose loss functions.

We believe that our taxonomy may help practitioners find the methods of choice that can be easily integrated into their existing tool chain. For instance, applying data transformations

can result in performance improvements while maintaining the current model architecture. Conversely, using specialized architectures, the data preprocessing pipeline can be kept intact.

IV. Deep Neural Networks for Tabular Data

In this section, we discuss the use of deep neural networks on tabular data for classification and regression tasks according to the taxonomy presented in Section III. We provide an overview of existing deep learning approaches in this area of research in Table II and examine the three methodological categories in detail: data transformation methods (see Section IV-A), architecture-based methods (see Section IV-B), and regularization-based models (see Section IV-C).

A. Data Transformation Methods

Most traditional approaches for deep neural networks on tabular data fall into this group. Interestingly, data preprocessing plays a relatively minor role in computer vision, even though the field is currently dominated by deep learning solutions [2]. There are many different possibilities to transform tabular data, and each may have a different impact on the learning results [47].

1) Single-Dimensional Encoding: One of the critical obstacles for deep learning with tabular data is categorical variables. Since neural networks only accept real number vectors as inputs, these values must be transformed before a model can use them. Therefore, the first class of methods attempts to encode categorical variables in a way suitable for deep learning models.

Approaches in this group [47] are divided into deterministic techniques, which can be used before training the model, and more complicated automatic techniques that are part of the model architecture. There are many ways for deterministic data encoding; hence, we restrict ourselves to the most common ones without the claim of completeness.

The simplest data encoding technique might be ordinal or label encoding. Every category is just mapped to a discrete numeric value, e.g., {`Apple`, `Banana`} are encoded as {0, 1}. One drawback of this method may be that it introduces an artificial order to previously unordered categories. Another straightforward method that does not induce any order is the one-hot encoding. One additional column for each unique category is added to the data. Only the column corresponding to the observed category is assigned the value one, with the other values being zero. In our example, `Apple` could be encoded as `(1,0)` and `Banana` as `(0,1)`. In the presence of a diverse set of categories in the data, this method can lead to high-dimensional sparse feature vectors and exacerbate the "curse of dimensionality" problem.

One approach that needs no extra columns and does not include any artificial order is the so-called leave-one-out encoding. It is based on the target encoding technique proposed in the work in [94], where every category is replaced with the mean of the target variable of that category. The leave-one-out encoding excludes the current row when computing the mean of the target variable to avoid overfitting. This approach is also used in the CatBoost framework [71], a state-of-the-art machine learning library for heterogeneous tabular data based on the gradient boosting algorithm [95].

A different strategy is hash-based encoding. Every category is transformed into a fixed-size value via a deterministic hash function. The output size is not directly dependent on the number of input categories but can be chosen manually.

2) Multidimensional Encoding: A first automatic encoding strategy is the value imputation and mask estimation (VIME) approach [79]. The authors propose a self-supervised and semisupervised deep learning framework for tabular data that trains an encoder in a self-supervised fashion by using two pretext tasks. Those tasks are independent of the concrete downstream task that the predictor has to solve. The first task of VIME is called mask vector estimation; its goal is to determine which values in a sample are corrupted. The second task, i.e., feature vector estimation, is to recover the original values of the sample. The encoder itself is a simple multilayer perceptron. This automatic encoding makes use of the fact that there is often much more unlabeled than labeled data. The encoder learns how to construct an informative homogeneous representation of the raw input data. In the semisupervised step, a predictive model, which is also a deep neural network model, is trained using the labeled and unlabeled data transformed by the encoder. For the encoder, a novel data augmentation method is used, corrupting an unlabeled data point multiple times with different masks. On the predictions from all augmented samples from one original data point, a consistency loss can be computed, which rewards similar outputs. To summarize, the VIME network trains an encoder, which is responsible to transform the categorical and numerical features into a new homogeneous and informative representation. This transformed feature vector is used as an input to the predictive model. For the encoder itself, the categorical data can be transformed by a simple one-hot encoding and binary encoding. The experimental results highlight how the self-supervised and semisupervised variants of the VIME framework can boost the performance over that of other baselines such as XGBoost. Even in the absence of unlabeled data, learning encodings in the proposed manner is shown to be beneficial for downstream performance.

Another stream of research aims at transforming the tabular input into a more homogeneous format. Since the revival of deep learning, convolutional neural networks have shown tremendous success in computer vision tasks. Therefore, Sun et al. [78] proposed the SuperTML method, which is a data conversion technique to transform tabular data into an image data format (2-D matrices), i.e., black-and-white images. On three datasets, SuperTML shows performance comparable with or superior to XGBoost.

The image generator for tabular data (IGTD) in [72] follows an idea similar to SuperTML. The IGTD framework converts tabular data into images to make use of classical convolutional architectures. As convolutional neural networks rely on spatial dependencies, the transformation into images is optimized by minimizing the difference between the feature distance ranking of the tabular data and the pixel distance ranking of the generated image. Every feature corresponds to one pixel,

TABLE II

OVERVIEW OF DEEP LEARNING APPROACHES FOR TABULAR DATA. WE ORGANIZE THEM IN CATEGORIES ORDERED CHRONOLOGICALLY INSIDE THE GROUPS. THE "INTERPRETABILITY" COLUMN INDICATES WHETHER THE APPROACH OFFERS SOME FORM INTERPRETABILITY FOR THE MODEL'S DECISIONS. THE KEY CHARACTERISTICS OF EVERY MODEL ARE SUMMARIZED IN THE LAST COLUMN

	Method	Interpretability	Key Characteristics
Encoding	SuperTML [78]		Transform tabular data into images for CNNs
	VIME [79]		Self-supervised learning and contextual embedding
	IGTD [72]		Transform tabular data into images for CNNs
	SCARF [80]		Self-supervised contrastive learning
Architectures, Hybrid	Wide&Deep [81]		Embedding layer for categorical features
	DeepFM [15]		Factorization machine for categorical data
	SDT [82]	✓	Distill neural network into interpretable decision tree
	xDeepFM [83]		Compressed interaction network
	TabNN [84]		DNNs based on feature groups distilled from GBDT
	DeepGBM [62]		Two DNNs, distill knowledge from decision tree
	NODE [7]		Differentiable oblivious decision trees ensemble
	NAM [85]	✓	Separate neural networks for each input variable
	NON [86]		Network-on-network model
	DNN2LR [87]		Calculate cross feature wields with DNNs for LR
	Net-DNF [50]		Structure based on disjunctive normal form
	Boost-GNN [88]		GNN on top decision trees from the GBDT algorithm
	SDTR [89]		Hierarchical differentiable neural regression model
Architectures, Transformer	TabNet [6]	✓	Sequential attention structure
	TabTransformer [90]	✓	Transformer network for categorical data
	SAINT [9]	✓	Attention over both rows and columns
	ARM-Net [91]		Adaptive relational modeling with multi-headgated attention network
	Non-Param. Transformer [92]		Process the entire data set at once, use attention between data points
Regul.	RLN [63]	✓	Hyperparameters regularization scheme
	STG [93]		Stochastic gate regularization
	Regularized DNNs [10]		A "cocktail" of regularization techniques

which leads to compact images with similar features close at neighboring pixels. Thus, IGDTs can be used in the absence of domain knowledge. The authors show relatively solid results for data with strong feature relationships, but the method may fail if the features are independent or feature similarities cannot characterize the relationships. In their experiments, the authors used only gene expression profiles and molecular descriptors of drugs as data. This kind of data may lead to a favorable inductive bias, so the general viability of the approach remains unclear.

B. Specialized Architectures

Specialized architectures form the largest group of approaches for deep tabular data learning. In this group, the focus is on the development and investigation of novel deep neural network architectures designed specifically for heterogeneous tabular data. Guided by the types of available models, we divide this group into two subgroups: hybrid models (presented in IV-B1) and transformer-based models (discussed in IV-B2).

1) Hybrid Models: Most approaches for deep neural networks on tabular data are hybrid models. They transform the data and fuse successful classical machine learning approaches, often decision trees, with neural networks. We distinguish between fully differentiable models, which can be differentiated with respect to all their parameters and partly differentiable models.

a) Fully differentiable models: The fully differentiable models in this category offer a valuable property: They permit end-to-end deep learning for training and inference by means of gradient descent optimizers. Thus, they allow for highly efficient implementations in modern deep learning frameworks that exploit GPU or TPU acceleration throughout the code.

Popov et al. [7] proposed an ensemble of differentiable oblivious decision trees [96]—also known as the NODE framework for deep learning on tabular data. Oblivious decision trees use the same splitting function for all nodes on the same level and can therefore be easily parallelized. NODE is inspired by the successful CatBoost [71] framework. To make the whole architecture fully differentiable and benefit from

end-to-end optimization, NODE utilizes the entmax transformation [97] and soft splits. In the original experiments, the NODE framework outperforms XGBoost and other GBDT models on many datasets. As NODE is based on decision tree ensembles, there is no preprocessing or transformation of the categorical data necessary. Decision trees are known to handle discrete features well. In the official implementation, strings are converted to integers using the leave-one-out encoding scheme. The NODE framework is widely used and provides a sound implementation that can be readily deployed.

Frosst and Hinton [82] contributed another model relying on soft decision trees (SDTs) to make neural networks more interpretable. They investigated training a deep neural network first, before using a mixture of its outputs and the ground-truth labels to train the SDT model in a second step. The authors showed that training a neural model first increases accuracy over SDTs that are directly learned from the data. However, their distilled trees still exhibit a performance gap to the neural networks that were fit in the initial step. Nevertheless, the model itself shows a clear relationship among different classes in a hierarchical fashion. It groups different categorical values based on the common patterns, e.g., digits 8 and 9 from the MNIST dataset [98]. To summarize, the proposed method allows for high interpretability and efficient inference, at the cost of slightly reduced accuracy.

Follow-up work [89] extends this line of research to heterogeneous tabular data and regression tasks and presents the SDT regressor (SDTR) framework. The SDTR is a neural network, which imitates a binary decision tree. Therefore, all neurons, such as nodes in a tree, get the same input from the data instead of the output from previous layers. In the case of deep networks, the SDTR could not beat other state-of-the-art models, but it has shown promising results in a low-memory setting, where single tree models and shallow architectures were compared.

Katzir et al. [50] followed the related idea. Their Net-DNF builds on the observation that every decision tree is merely a form of a Boolean formula, more precisely a disjunctive normal form. They use this inductive bias to design the architecture of a neural network, which is able to imitate the characteristics of the GBDT algorithm. The resulting Net-DNF was tested for classification tasks on datasets with no missing values, where it showed the results that are comparable to those of XGBoost [46]. However, the authors did not mention how to handle high-cardinality categorical data, as the used datasets contained mostly numerical and few binary features.

Linear models (e.g., linear and logistic regression) provide global interpretability but are inferior to complex deep neural networks. Usually, handcrafted feature engineering is required to improve the accuracy of linear models. Liu et al. [87] used a deep neural network to combine the features in a possibly nonlinear way; the resulting combination of features then serves as input to the linear model. In their approach—termed DDN2LR—this enhances the simple, interpretable linear model. In experimental evaluations, DNN2LR can outperform other more complex DNN models while maintaining some extent of interpretability.

The work by Cheng et al. [81] proposes a hybrid architecture that consists of linear and deep neural network models—Wide&Deep. A linear model that takes single features and a wide selection of handcrafted logical expressions on features as an input is enhanced by a deep neural network to improve the generalization capabilities. In addition, Wide&Deep learns an n-dimensional embedding vector for each categorical feature. All embeddings are concatenated resulting in a dense vector used as input to the neural network. The final prediction can be understood as a sum of both models. Experiments with a real-world system for app recommendation confirmed that users installed apps suggested by Wide&Deep were significantly more often than those provided by the previous model. A similar work by Guo and Berkhahn [99] proposes an embedding using deep neural networks for categorical variables.

Another contribution to the realm of Wide&Deep models is DeepFM [15]. The authors demonstrate that it is possible to replace the handcrafted feature transformations with learned factorization machines (FMs) [100]. The FM is an extension of a linear model designed to capture lower order interactions between features within high-dimensional and sparse data efficiently. Higher order interactions are modeled by a deep neural network. Similar to the original Wide&Deep model, DeepFM also relies on the same embedding vectors for its "wide" and "deep" parts. In contrast to the original Wide&Deep model, however, DeepFM alleviates the need for manual feature engineering. The experimental results show a solid improvement in CTR prediction tasks compared to a variety of models relying on either low- or high-order dependencies only and compared to other hybrid approaches.

Finally, network-on-network (NON) [86] is a classification model for tabular data, which focuses on capturing the intrafeature information efficiently. It consists of three components: a fieldwise network consisting of one unique deep neural network for every column to capture the column-specific information, an across-field network, which chooses the optimal operations based on the dataset, and an operation fusion network, connecting the chosen operations allowing for nonlinearities. As the optimal operations for the specific data are selected, the performance is considerably better than that of other deep learning models. However, decision trees, the current state-of-the-art models for tabular data, were not listed among the baselines. Also, training as many neural networks as columns and selecting the operations on the fly may lead to a long computation time.

b) Partly differentiable models: This subgroup of hybrid models aims at combining nondifferentiable approaches with deep neural networks. Models from this group usually utilize decision trees for the nondifferentiable part.

The DeepGBM model [62] combines the flexibility of deep neural networks with the preprocessing capabilities of GBDTs. DeepGBM consists of two neural networks—CatNN and GBDT2NN. While CatNN is specialized to handle sparse categorical features, GBDT2NN is specialized to deal with dense numerical features.

In the preprocessing step for the CatNN network, the categorical data are transformed via ordinal encoding (to convert

the potential strings into integers), and the numerical features are discretized, as this network is specialized for categorical data. The GBDT2NN network distills the knowledge about the underlying dataset from a model based on GBDTs by accessing the leaf indices of the decision trees. This embedding based on decision tree leaves was first proposed in [101] for the random forest algorithm. Later, the same knowledge distillation strategy has been adopted for GBDTs [102].

Using the proposed combination of two deep neural networks, DeepGBM has a strong learning capacity for both categorical and numerical features. Distinctively, the authors implemented and tested DeepGBM's online prediction performance, which is significantly higher than that of GBDTs. On the downside, the leaf indices can be seen as meta categorical features since these numbers cannot be directly compared. Also, it is not clear how other data-related issues, such as missing values, different scaling of numeric features, and noise influence the predictions produced by the models.

The TabNN architecture, introduced by Ke et al. [84], is based on two principles: explicitly leveraging expressive feature combinations and reducing model complexity. It distills the knowledge from GBDTs to retrieve feature groups; it clusters them and then constructs the neural network based on those feature combinations. Also, structural knowledge from the trees is transferred to provide an effective initialization. The experimental results show that the performance of a GBDT model can be further improved by leveraging its feature sets in combination with neural encoders. Furthermore, TabNN shows promising results on streaming data. However, the construction of the network already takes different extensive computation steps of which one is only a heuristic to avoid an NP-hard problem. Unfortunately, these computational challenges and the unavailability of an implementation limit the practical usability of the network.

In similar spirit to DeepGBM and TabNN, the work by Ivanov and Prokhorenkova [88] proposed using GBDTs for the data prepossessing step. They exploited the fact that decision trees are special cases of directed graphs and process decision trees using graph neural networks. Thus, the proposed framework exploits the topology information from the decision trees using graph neural networks [103]. The resulting architecture is coined boosted graph neural network (BGNN). In multiple experiments, BGNN demonstrates that the proposed architecture is superior to other state-of-the-art graph neural networks in terms of predictive performance and training time and also outperforms GDBT models on most of the datasets.

2) Transformer-Based Models: Transformer-based approaches form another subgroup of model-based deep neural methods for tabular data. Inspired by the recent surge of interest in transformer-based methods and their successes on text and visual data [66], [104], researchers and practitioners have proposed multiple approaches using deep attention mechanisms [5] for heterogeneous tabular data.

TabNet [6] is one of the first transformer-based models for tabular data. Like a decision tree, the TabNet architecture comprises multiple subnetworks that are processed in a sequential hierarchical manner. According to [6], each subnetwork corresponds to one decision step. To train TabNet, each decision step (subnetwork) receives the current data batch as input. TabNet aggregates the outputs of all decision steps to obtain the final prediction. At each decision step, TabNet first applies a sparse feature mask [105] to perform soft instancewise feature selection. The authors claim that the feature selection can save valuable resources, as the network may focus on the most important features. The feature mask of a decision step is trained using attentive information from the previous decision step. To this end, a feature transformer module decides which features should be passed to the next decision step and which features should be used to obtain the output at the current decision step. Some layers of the feature transformers are shared across all decision steps. The obtained feature masks correspond to local feature weights and can also be combined into a global importance score. Accordingly, TabNet is one of the few deep neural networks that offers different levels of interpretability by design. Indeed, experiments show that each decision step of TabNet tends to focus on a particular subdomain of the learning problem (i.e., one particular subset of features). This behavior is similar to convolutional neural networks. TabNet also provides a decoder module that is able to preprocess input data (e.g., replace missing values) in an unsupervised way. Accordingly, TabNet can be used in a two-stage self-supervised learning procedure, which improves the overall predictive quality. The experiments confirm the improved feature selection process, which leads to smaller models with less trainable parameters. Also, TabNet outperforms tree-based models and MLPs consistently while providing a more accurate interpretation of the feature importance. One of the popular Python [106] frameworks for tabular data provides an efficient implementation of TabNet [107]. Recently, TabNet has also been investigated in the context of fair machine learning [108], [109]. Attention-based architectures offer mechanisms for interpretability, which is an essential advantage over many hybrid models. Fig. 2 shows attention maps of the TabNet model and KernelSHAP explanation framework on the Adult dataset [54].

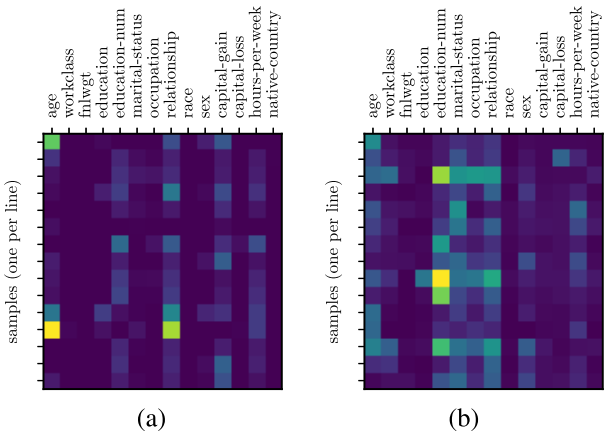

Fig. 2. Interpretable learning with the TabNet [6] architecture. We compare the attributions provided by the model for a sample from the UCI Adult dataset with those provided by the game theoretic KernelSHAP framework [116]. (a) TabNet attributions. (b) KernelSHAP attributions.

Another supervised and semisupervised approach is introduced by Huang et al. [90]. Their TabTransformer architecture

uses self-attention-based transformers to map the categorical features to contextual embedding. This embedding is more robust to missing or noisy data and enables interpretability. The embedded categorical features are then together with the numerical ones fed into a simple multilayer perceptron. If, in addition, there is an extra amount of unlabeled data, unsupervised pretraining can improve the results, using masked language modeling or replacing token detection. Extensive experiments show that TabTransformer matches the performance of tree-based ensemble techniques, showing success also when dealing with missing or noisy data. The TabTransformer network puts a significant focus on the categorical features. It transforms the embedding of those features into contextual embedding, which is then used as input for the multilayer perceptron. This embedding is implemented by different multihead attention-based transformers, which are optimized during training.

ARM-net [91] is an adaptive neural network for relation modeling tailored to tabular data. The key idea of the ARM-net framework is to model feature interactions with combined features (feature crosses) selectively and dynamically by first transforming the input features into exponential space and then determining the interaction order and interaction weights adaptively for each feature cross. Furthermore, the authors propose a novel sparse attention mechanism to generate the interaction weights given the input data dynamically. Thus, users can explicitly model feature crosses of arbitrary orders with noisy features filtered selectively. On five real-world datasets, ARM-net shows its superior effectiveness in representing feature interactions compared to various baselines, which model the feature interactions in different ways.

Self-attention and intersample attention transformer (SAINT) [9] is a hybrid attention approach, combining self-attention [5] with intersample attention over multiple rows. When handling missing or noisy data, this mechanism allows the model to borrow the corresponding information from similar samples, which improves the model's robustness. The technique is reminiscent of nearest neighbor imputation. In addition, all features are embedded into a combined dense latent vector, enhancing existing correlations between values from one data point. To exploit the presence of unlabeled data, a self-supervised contrastive pre-training can further improve the results, minimizing the distance between two views of the same sample and maximizing the distance between different ones. Like the VIME framework (Section IV-A1), SAINT uses CutMix [110] to augment samples in the input space and uses mixup [111] in the embedding space. The experimental results show that SAINT outperforms tree-based models like XGBoost as well as other deep learning approaches for tabular data on average. When unlabeled data are available, the performance can be improved further using the proposed pretraining.

Finally, even some new learning paradigms are being proposed. For instance, the nonparametric transformer (NPT) [92] does not construct a mapping from individual inputs to outputs but uses the entire dataset at once. By using attention between data points, relations between arbitrary samples can be modeled and leveraged for classifying test samples. Experiments confirmed that this new approach can reach state-of-the-art results on most datasets by using intersample attention mechanisms.

C. Regularization Models

The third group of approaches argues that extreme flexibility of deep learning models for tabular data is one of the main learning obstacles and strong regularization of learned parameters may improve the overall performance.

One of the first methods in this category was the regularization learning network (RLN) proposed by Shavitt and Segal [63], which uses a learned regularization scheme. The main idea is based on the observation that features in tabular datasets have very different importances. Contrarily to other data modalities data such as images or text, a single tabular feature may change the entire prediction. Therefore, the authors apply trainable regularization coefficients to each single weight in a neural network, hence allowing high sensitivity with respect to certain inputs or network parts while being insensitive to others. To efficiently determine the corresponding coefficients, the authors propose a novel loss function termed "counterfactual loss." The regularization coefficients lead to a very sparse network, which also provides the importance of the remaining input features.

In their experiments, RLNs outperform deep neural networks and obtain the results comparable to those of the GBDT algorithm, but the evaluation relies on a dataset with mainly numerical data to compare the models. The RLN paper does not address the issues of categorical data. For the experiments and the example implementation, datasets with exclusively numerical data (except for the gender attribute) were used. A similar idea is proposed in [112], where regularization coefficients are learned only in the first layer with a goal to extract feature importance.

Kadra et al. [10] stated that simple multilayer perceptrons can outperform state-of-the-art algorithms on tabular data if deep learning networks are properly regularized. The authors propose a "cocktail" of regularization with 13 different techniques that are applied jointly. From those, the optimal subset and their subsidiary hyperparameters are selected. They demonstrate in extensive experiments that the regularization "cocktails" can not only improve the performance of multilayer perceptrons but these simple models also outperform tree-based architectures. On the downside, the extensive per-dataset regularization and hyperparameter optimization take much more computation time than the GBDT algorithm.

There are several other noteworthy works [113], [114], [115], indicating that strong regularization of deep neural networks can be beneficial for tabular data.

V. Tabular Data Generation

For many applications, the generation of realistic tabular data is fundamental. Three of the main purposes are data augmentation [117], data imputation (i.e., the filling of missing values) [118], [119], and rebalancing [36], [37], [120], [121]. Another highly relevant topic is privacy-aware machine learning [38], [39], [122] where generated data can potentially be leveraged to overcome privacy concerns.

A. Methods

While the generation of images and text is highly explored [123], [124], [125], generating synthetic tabular data is a less frequent concern. The mixed structure of discrete and continuous features along with their different value distributions still poses a significant challenge.

Classical approaches for the data generation task include Copulas [126], [127] and Bayesian networks [128]. Among Bayesian networks, those based on the Chow–Liu approximation [129] are especially popular [38], [130], [131], [132].

In the deep learning era, generative adversarial networks (GANs) [133] have proven highly successful for the generation of images [123], [134]. GANs were recently introduced as an original way to train a generative deep neural network model. They consist of two separate models: a generator G that generates samples from the data distribution and a discriminator D that estimates the probability that a sample came from the ground-truth distribution. Both G and D are usually chosen to be nonlinear functions such as multilayer perceptrons. To learn a generator distribution p_g over data $\boldsymbol{x}$, the generator $G(\boldsymbol{z}; \theta_g)$ maps the samples from a noise distribution $p_z(\boldsymbol{z})$ (e.g., the Gaussian distribution) to the input data space. The discriminator $D(\boldsymbol{x}; \theta_d)$ outputs the probability that a data point $\boldsymbol{x}$ comes from the training data's distribution p_{data} rather than from the generator's output distribution p_g. During joint training of G and D, G will start generating successively more realistic samples to fool the discriminator D. For more details on GANs, we refer the interested reader to the original paper [133].

In Table III, we provide an overview of tabular generation approaches that use deep learning techniques. Note that due to the enormous number of approaches, we list the most influential works that address the problem of data generation with a particular focus on tabular data. We exclude works that are targeted toward highly domain-specific tasks.

Although it was found that GANs lag behind at the generation of discrete outputs such as natural language [125], they are still frequently chosen to generate tabular data. Vanilla GANs or derivates, such as the Wasserstein GAN (WGAN) [135], WGAN with gradient penalty (WGAN-GP) [136], Cramér GAN [137], or the Boundary seeking GAN [138], which is designed to model discrete data, are commonly used in the literature to generate tabular data (cf. Table III). Moreover, VeeGAN [139] is frequently used as a reference for tabular data generation [38], [130], [131]. Apart from GANs, autoencoder-based architectures—in particular those relying on variational autoencoders (VAEs) [140]—have been proposed [130], [141].

In the following, we will briefly discuss the most relevant approaches that helped shape the domain. For example, MedGAN [39] was one of the first works and provides a deep learning model to generate patient records. As all the features in their work are discrete, this model cannot be easily transferred to arbitrary tabular datasets. The table-GAN approach in [142] adapts the deep convolutional GAN for tabular data. Specifically, the features from one record are converted into a matrix so that they can be processed by convolutional filters of a convolutional neural network. However, it remains unclear to which extent the inductive bias used for images are suitable for tabular data.

The approach by Xu et al. [130] focuses on the correlation between the features of one data point. The authors first propose the mode-specific normalization technique for data preprocessing that allows to transform non-Gaussian distributions in the continuous columns. They express numeric values in terms of a mixture component number and the deviation from that component's center. This allows to represent multimodal and skewed distributions. Their generative solution, coined CTGAN, uses the conditional GAN architecture to enforce learning proper conditional distributions for each column. To obtain categorical values and to allow for backpropagation in the presence of categorical values, the gumbel-softmax trick [143] is utilized. The authors also propose a model based on VAEs, named tabular VAE (TVAE), which outperforms their suggested GAN approach. Both approaches can be considered state of the art.

While GANs and VAEs are prevalent, other recently proposed architectures include machine-learned causal models [144] and invertible flows [38]. When privacy is the main factor of concern, models, such as PATE-GAN [145], provide generative models with certain differential privacy guarantees. Although very relevant for practical applications, such privacy guarantees and related federated learning approaches with tabular data [146] are outside the scope of this review.

Fan et al. [122] compared a variety of different GAN architectures for tabular data synthesis and recommended using a simple, fully connected architecture with a vanilla GAN loss with minor changes to prevent mode collapse. They also use the normalization proposed in [130]. In their experiments, the WGAN loss or the use of convolutional architectures on tabular data does boost the generative performance.

B. Assessing Generative Quality

To assess the quality of the generated data, several performance measures are used. The most common approach is to define a proxy classification task and train one model for it on the real training set and another on the artificially generated dataset. With a highly capable generator, the predictive performance of the artificial-data model on the real-data test set should be almost on par with its real-data counterpart. This measure is often referred to as machine learning efficacy and used in [39], [131], and [147]. In nonobvious classification tasks, an arbitrary feature can be used as a label and predicted [39], [148], [149]. Another approach is to visually inspect the modeled distributions per feature, e.g., the cumulative distribution functions [117], or compare the expected values in scatter plots [39], [148]. A more quantitative approach is the use of statistical tests, such as the Kolmogorov–Smirnov test [152], to assess the distributional difference [149]. On synthetic datasets, the output distribution can be compared to the ground truth, e.g., in terms of log likelihood [130], [144]. Because overfitted models can also obtain good scores, Xu et al. [130] proposed evaluating the likelihood of a test set under an estimate of the GAN's output distribution. Especially in a privacy-preserving context,

TABLE III
GENERATION OF TABULAR DATA USING DEEP NEURAL NETWORK MODELS (IN CHRONOLOGICAL ORDER)

Method	Based upon	Application
medGAN, medWGAN [39]	Autoencoder+GAN	Medical Records
TableGAN [142]	DCGAN	General
Mottini et al. [147]	Cramér GAN	Passenger Records
Camino et al. [148]	medGAN, ARAE	General
medBGAN, medWGAN [149]	WGAN-GP, Boundary seeking GAN	Medical Records
ITS-GAN [117]	GAN with AE for constraints	General
CTGAN, TVAE [130]	Wasserstein GAN, VAE	General
artGAN [121]	WGAN-GP	Health Data
VAEM [141]	VAE (Hierarchical)	General
OVAE [131]	Oblivious VAE	General
TAEI [37]	AE+SMOTE (in multiple setups)	General
Causal-TGAN [150]	Causal-Model, WGAN-GP	General
Copula-Flow [38]	Invertible Flows	General
Synthsonic [132]	Copula + CLBNs	General
GReaT [151]	Language Transformer	General

the distribution of the distance to closest record (DCR) can be calculated and compared to the respective distances on the test set [142]. This measure is important to assess the extent of sample memorization. Overall, we conclude that a single measure is not sufficient to assess the generative quality. For instance, a generative model that memorizes the original samples will score well in the machine learning efficiency metric but fail the DCR check. Therefore, we highly recommend using several evaluation measures that focus on individual aspects of data quality.

VI. EXPLANATION MECHANISMS FOR DEEP LEARNING WITH TABULAR DATA

Explainable machine learning is concerned with the problem of providing explanations for complex machine learning models. With stricter regulations for automated decision-making [41] and the adoption of machine learning models in high-stakes domains such as finance and healthcare [45], [153], [154], interpretability is becoming a key concern. Toward this goal, various streams of research follow different explainability paradigms. Among these, feature attribution methods and counterfactual explanations are two of the popular forms [155], [156], [157]. Because these techniques are gaining importance for researchers and practitioners alike, we dedicate the following to reviewing these methods.

A. Feature Highlighting Explanations

Local input attribution techniques seek to explain the behavior of machine learning models instance by instance. Those methods aim to highlight the influence of the inputs that have on the prediction by assigning importance scores to the input features. Some popular approaches for model explanations aim at constructing classification models that are explainable by design [158], [159], [160]. This is often achieved by enforcing the deep neural network model to be locally linear. Moreover, if the model's parameters are known and can be accessed, then the explanation technique can use these parameters to generate the model explanation. For such settings, relevance-propagation-based methods, e.g., [161], [162], and gradient-based approaches, e.g., [163], [164], [165], have been suggested. In cases where the parameters of the neural network cannot be accessed, model-agnostic approaches can prove useful. This group of approaches seeks to explain a model's behavior locally by applying surrogate models [116], [166], [167], [168], [169], which are interpretable by design and are used to explain individual predictions of black-box machine learning models. In order to test the performance of these black-box explanations techniques, Liu et al. [170] suggested a python-based benchmarking library.

B. Counterfactual Explanations

From the perspective of algorithmic recourse, the main purpose of counterfactual explanations is to suggest constructive interventions to the input of a deep neural network so that the output changes to the advantage of an end user. In simple terms, a minimal change to the feature vector that will flip the classification outcome is computed and provided as an explanation. By emphasizing both the feature importance and the recommendation aspect, counterfactual explanation methods can be further divided into three different groups: works that assume that all features can be independently manipulated [171] and works that focus on manifold constraints to capture feature dependencies.

In the class of independence-based methods, where the input features of the predictive model are assumed to be independent, some approaches use combinatorial solvers to generate recourse in the presence of feasibility constraints [172], [173], [174], [175]. Another line of research deploys gradient-based optimization to find low-cost counterfactual explanations in the presence of feasibility and diversity constraints [176], [177]. The main problem with these approaches is that they abstract from input correlations.

To alleviate this problem and to suggest realistic-looking counterfactuals, researchers have suggested building recourse suggestions on generative models [178], [179], [180], [181], [182]. The main idea is to change the geometry of the intervention space to a lower dimensional latent space, which encodes different factors of variation while capturing input dependencies. To this end, these methods primarily use (tabular data) VAEs [140], [183]. In particular, Mahajan et al. [181] demonstrated how to encode various feasibility constraints into such models. However, an extensive comparison across this class of methods is still missing since it is difficult to measure how realistic the generated data are in the context of algorithmic recourse.

More recently, a few works have suggested to develop counterfactual explanations that are robust to model shifts

and noise in the recourse implementations [184], [185], [186]. A comprehensive treatment on how to extend these lines of work to arbitrary high-cardinality categorical variables is still an open problem in the field.

For a more fine-grained overview over the literature on counterfactual explanations, we refer the interested reader to the most recent surveys [187], [188]. Finally, Pawelczyk et al. [157] implemented an open-source python library, which provides support for many of the aforementioned counterfactual explanation models.

VII. Experiments

Although several experimental studies have been published in recent years [8], [10], an exhaustive comparison between existing deep learning approaches for heterogeneous tabular data is still missing in the literature. For example, important aspects of deep learning models, such as training and inference time, model size, and interpretability, are not discussed.

To fill this gap, we present an extensive empirical comparison of machine and deep learning methods on real-world datasets with varying characteristics in this section. We discuss the dataset choice (VII-A), the results (VII-B), and present a comparison of the training and inference time for all the machine learning models considered in this survey (VII-C). We also discuss the size of deep learning models. Finally, to the best of our knowledge, we present the first comparison of explainable deep learning methods for tabular data (VII-D). We release the full source code of our experiments for maximum transparency.[1]

A. Datasets

In computer vision, there are many established datasets for the evaluation of new deep learning architectures such as MNIST [98], CIFAR [189], and ImageNet [190]. On the contrary, there are no established standard heterogeneous datasets. Carefully checking the works listed in Section IV, we identified over 100 different datasets with different characteristics in their respective experimental evaluation sections. We note that the small overlap between the mentioned works makes it hard to compare the results across these works in general. Therefore, in this work, we deliberately select datasets covering the entire range of characteristics, such as data domain (e.g., finance, e-commerce, geography, and physics), different types of target variables (classification and regression), varying number of categorical variables and continuous variables, and differing sample sizes (small to large). Furthermore, most of the selected datasets were previously featured in multiple studies.

The first dataset of our study is the Home Equity Line of Credit (HELOC) dataset provided by FICO [191]. This dataset consists of anonymized information from real homeowners who applied for home equity lines of credit. An HELOC is a line of credit typically offered by a bank as a percentage of home equity. The task consists of using the information about the applicant in their credit report to predict whether they will repay their HELOC account within a two-year period.

We further use the Adult Income dataset [54], which is among the most popular tabular datasets used in the surveyed work (five usages). It includes basic information about individuals such as age, gender, and education. The target variable is binary; it represents high and low income.

The largest tabular dataset in our study is HIGGS, which stems from particle physics. The task is to distinguish between signals with Higgs bosons (HIGGS) and a background process [192]. Monte Carlo simulations [193] were used to produce the data. In the first 21 columns (columns 2-22), the particle detectors in the accelerator measure kinematic properties. In the last seven columns, these properties are analyzed. In total, HIGGS includes 11 million rows. We also binarize the 21st variable into a categorical variable with three groups since DeepFM, DeepGBM, TabTransformer, and SAINT models require at least one categorical attribute, to benchmark the method's special functionality on large datasets.

The Covertype dataset [54] is multiclassification dataset, which holds cartographic information about land cells (e.g., elevation and slope). The goal is to predict which one out of seven forest cover types is present in the cell.

Finally, we utilize the California Housing dataset [194], which contains information about a number of properties. The prediction task (regression) is to estimate the price of the corresponding home.

The fundamental characteristics of the selected datasets are summarized in Table IV.

TABLE IV
Main Properties of the Real-World Heterogeneous Tabular Datasets Used in This Survey. We Also Indicate the Dataset Task, Where "Binary" Stands for Binary Classification and "Multi-Class" Represents Multiclass Classification

	HELOC	Adult Income	HIGGS	Covertype	California Housing
#Samples	9.871	32.561	11 M.	581.012	20.640
#Num. features	21	6	27	52	8
#Cat. features	2	8	1	2	0
Task	Binary	Binary	Binary	Multi-Class	Regression
#Classes	2	2	2	7	-

B. Open Performance Benchmark on Tabular Data

1) Hyperparameter Selection: In order to do a fair evaluation, we use the Optuna library [199] with 100 iterations for each model to tune hyperparameters. Each hyperparameter configuration was cross-validated with five folds. The hyperparameter ranges used are publicly available online along with our code. We laid out the search space based on the information given in the corresponding papers and recommendations from the framework's authors.

2) Data Preprocessing: We prepossessed the data in the same way for every machine learning model by applying zero-mean, unit-variance normalization to the numerical features and an ordinal encoding to the categorical ones using the

[1]Open benchmarking on tabular data for machine learning models: https://github.com/kathrinse/TabSurvey.

TABLE V

OPEN PERFORMANCE BENCHMARK RESULTS BASED ON (STRATIFIED) FIVEFOLD CROSS VALIDATION. WE USE THE SAME FOLD SPLITTING STRATEGY FOR EVERY DATASET. THE TOP RESULTS FOR EACH DATASET ARE IN **BOLD**, WE ALSO UNDERLINE THE SECOND-BEST RESULTS. THE MEAN AND STANDARD DEVIATION VALUES ARE REPORTED FOR EACH BASELINE MODEL. MISSING RESULTS INDICATE THAT THE CORRESPONDING MODEL COULD NOT BE APPLIED TO THE TASK TYPE (REGRESSION OR MULTICLASS CLASSIFICATION)

	Method	HELOC		Adult		HIGGS		Covertype		Cal. Housing
		Acc ↑	AUC ↑	Acc ↑	AUC ↑	Acc ↑	AUC ↑	Acc ↑	AUC ↑	MSE ↓
Machine Learning	Linear Model	73.0±0.0	80.1±0.1	82.5±0.2	85.4±0.2	64.1±0.0	68.4±0.0	72.4±0.0	92.8±0.0	0.528±0.008
	KNN [58]	72.2±0.0	79.0±0.1	83.2±0.2	87.5±0.2	62.3±0.1	67.1±0.0	70.2±0.1	90.1±0.2	0.421±0.009
	Decision Trees [195]	80.3±0.0	89.3±0.1	85.3±0.2	89.8±0.1	71.3±0.0	78.7±0.0	79.1±0.0	95.0±0.0	0.404±0.007
	Random Forest [196]	82.1±0.2	90.0±0.2	86.1±0.2	91.7±0.2	71.9±0.0	79.7±0.0	78.1±0.1	96.1±0.0	0.272±0.006
	XGBoost [46]	83.5±0.2	92.2±0.0	87.3±0.2	92.8±0.1	77.6±0.0	85.9±0.0	**97.3±0.0**	**99.9±0.0**	0.206±0.005
	LightGBM [70]	83.5±0.1	92.3±0.0	**87.4±0.2**	**92.9±0.1**	77.1±0.0	85.5±0.0	93.5±0.0	99.7±0.0	**0.195±0.005**
	CatBoost [71]	**83.6±0.3**	**92.4±0.1**	87.2±0.2	92.8±0.1	77.5±0.0	85.8±0.0	96.4±0.0	99.8±0.0	0.196±0.004
	Model Trees [197]	82.6±0.2	91.5±0.0	85.0±0.2	90.4±0.1	69.8±0.0	76.7±0.0	-	-	0.385±0.019
Deep Learning	MLP [198]	73.2±0.3	80.3±0.1	84.8±0.1	90.3±0.2	77.1±0.0	85.6±0.0	91.0±0.4	76.1±3.0	0.263±0.008
	VIME [79]	72.7±0.0	79.2±0.0	84.8±0.2	90.5±0.2	76.9±0.2	85.5±0.1	90.9±0.1	82.9±0.7	0.275±0.007
	DeepFM [15]	73.6±0.2	80.4±0.1	86.1±0.2	91.7±0.1	76.9±0.0	83.4±0.0	-	-	0.260±0.006
	DeepGBM [62]	78.0±0.4	84.1±0.1	84.6±0.3	90.8±0.1	74.5±0.0	83.0±0.0	-	-	0.856±0.065
	NODE [7]	79.8±0.2	87.5±0.2	85.6±0.3	91.1±0.2	76.9±0.1	85.4±0.1	89.9±0.1	98.7±0.0	0.276±0.005
	NAM [85]	73.3±0.1	80.7±0.3	83.4±0.1	86.6±0.1	53.9±0.6	55.0±1.2	-	-	0.725±0.022
	Net-DNF [50]	82.6±0.4	91.5±0.2	85.7±0.2	91.3±0.1	76.6±0.1	85.1±0.1	94.2±0.1	99.1±0.0	-
	TabNet [6]	81.0±0.1	90.0±0.1	85.4±0.2	91.1±0.1	76.5±1.3	84.9±1.4	93.1±0.2	99.4±0.0	0.346±0.007
	TabTransformer [90]	73.3±0.1	80.1±0.2	85.2±0.2	90.6±0.2	73.8±0.0	81.9±0.0	76.5±0.3	72.9±2.3	0.451±0.014
	SAINT [9]	82.1±0.3	90.7±0.2	86.1±0.3	91.6±0.2	**79.8±0.0**	**88.3±0.0**	96.3±0.1	99.8±0.0	0.226±0.004
	RLN [63]	73.2±0.4	80.1±0.4	81.0±1.6	75.9±8.2	71.8±0.2	79.4±0.2	77.2±1.5	92.0±0.9	0.348±0.013
	STG [93]	73.1±0.1	80.0±0.1	85.4±0.1	90.9±0.1	73.9±0.1	81.9±0.1	81.8±0.3	96.2±0.0	0.285±0.006

alphabetical order. According to Hancock and Khoshgoftaar [47], the chosen encoding strategy shows comparable performance to more advanced methods. The missing values were substituted with zeros for the linear regression and models based on pure neural networks since these methods cannot accept them otherwise. We explicitly specify categorical features for LightGBM, DeepFM, DeepGBM, TabNet, TabTransformer, and SAINT since these approaches provide special functionality dedicated to categorical values, e.g., learning an embedding of the categories. As we noted in Section III-C, the results of experiments may be affected by the data preprocessing.

3) Reproducibility and Extensibility: For maximum reproducibility, we run all experiments in a docker container [200]. We underline again that our full code is publicly released so that the experiments can be replicated. The mentioned datasets are also publicly available and can be used as a benchmark for novel methods. We would highly welcome contributed implementations of additional methods from the data science community.

4) Results: The results of our experiments are shown in Table V. They draw a different picture than many recent research papers may suggest: for all but the very large HIGGS dataset, the best scores are still obtained by boosted decision tree ensembles. XGBoost and CatBoost outperform all deep learning-based approaches on the small and medium datasets, the regression dataset, and the multiclass dataset. For the large-scale HIGGS, SAINT outperforms the classical machine learning approaches. This suggests that for very large tabular datasets with predominantly continuous features, modern neural network architectures may have an advantage over classical approaches after all. In general, however, our results are consistent with the inferior performance of deep learning techniques in comparison to approaches based on decision tree ensembles (such as GBDT) on tabular data that were observed in various Kaggle competitions [201].

Considering only deep learning approaches, we observe that SAINT provided competitive results across datasets. However, for the other models, the performance was highly dependent on the chosen dataset. DeepFM performed best (among the deep learning models) on the Adult dataset and second-best on the California Housing dataset, but returned only weak results on the HELOC dataset.

C. Run Time Comparison

We also analyze the training and inference time of the models in comparison to their performance. We plot the time–performance characteristic for the models in Figs. 3 and 4 for the Adult and the HIGGS dataset, respectively. While the training time of gradient boosting-based models is lower than that of most deep neural network-based methods, their inference time on the HIGGS dataset with 11 million samples is significantly higher: for XGBoost, the inference time amounts to 5995 s, whereas inference times for MLP and SAINT are 10.18 and 282 s, respectively. All

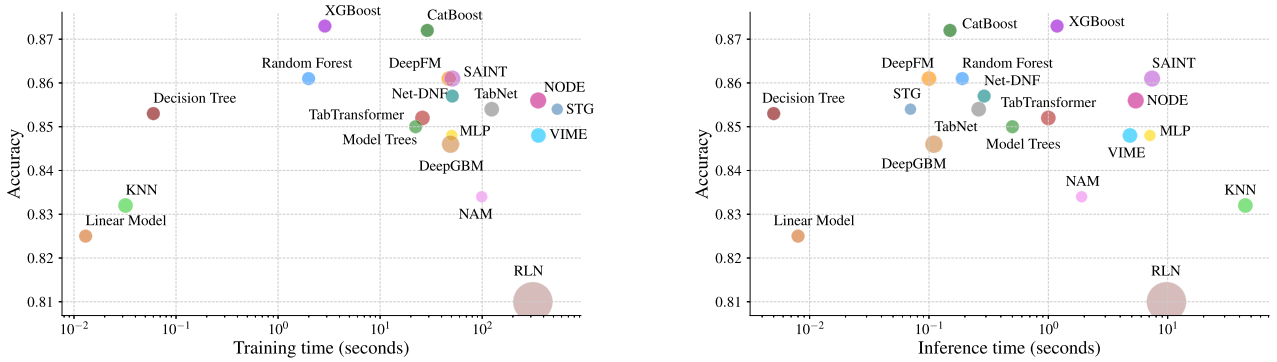

Fig. 3. Train (left) and inference (right) time benchmarks for selected methods on the Adult dataset with 32.561 samples. The circle size reflects the accuracy standard deviation.

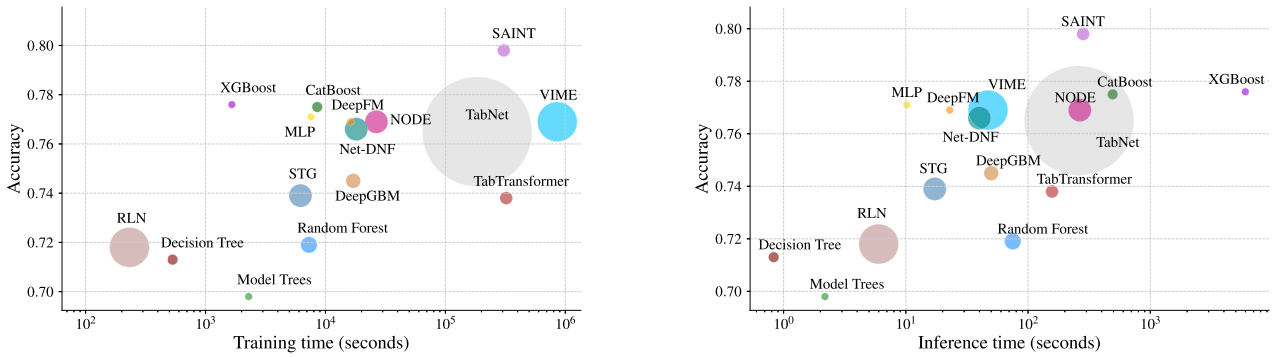

Fig. 4. Train (left) and inference (right) time benchmarks for selected methods on the HIGGS dataset with 11 million samples. The circle size reflects the accuracy standard deviation.

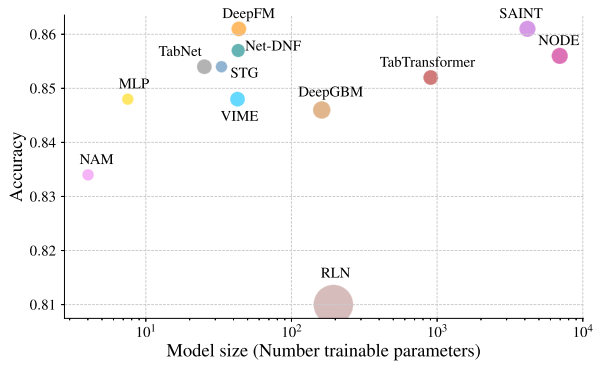

Fig. 5. Size comparison of deep learning models on the Adult dataset. The circle size reflects the standard deviation.

gradient boosting and deep learning models were trained on the same GPU.

D. Interpretability Assessment

As opposed to the pure on-task performance, interpretability of the models is becoming an increasingly important characteristic. Therefore, we end this section with a distinct assessment of the interpretability properties claimed by some methods. The model size (number of parameters) can provide a first intuition of the interpretability of the models. Therefore, we provide a size comparison of deep learning models in Fig. 5.

Admittedly, explanations can be provided in very different forms, which may each have their own use cases. Hence, we can only compare explanations that have a common form. In this work, we chose feature attributions as the explanation format because they are the prevalent form of post hoc explainability for the models considered in this work. Remarkably, the models that build on the transformer architecture (Section IV-B2) often claim some extent of interpretability through the attention maps [9]. To verify this hypothesis and assess the attribution provided by some of the frameworks in practice, we run an ablation test with the features that were attributed the highest importance over all samples. Furthermore, due to the lack of ground-truth attribution values, we compare individual attributions to the well-known KernelSHAP values [116].

Evaluation of the quality of feature attribution is known to be a nontrivial problem [202]. We measure the fidelity [203] of the attributions by successively removing the features that have the highest mean importance assigned (most relevant first (MoRF) [203]). We then retrain the model on the reduced feature set. A sharp drop in predictive accuracy indicates that the discriminative features were successfully identified and removed. We do the same for the inverse order, least relevant first (LeRF), which removes the features deemed unimportant. In this case, the accuracy should stay high as long as possible. For the attention maps of TabTransformer and SAINT, we either use the sum over the entire columns of the intrafeature attention maps as an importance

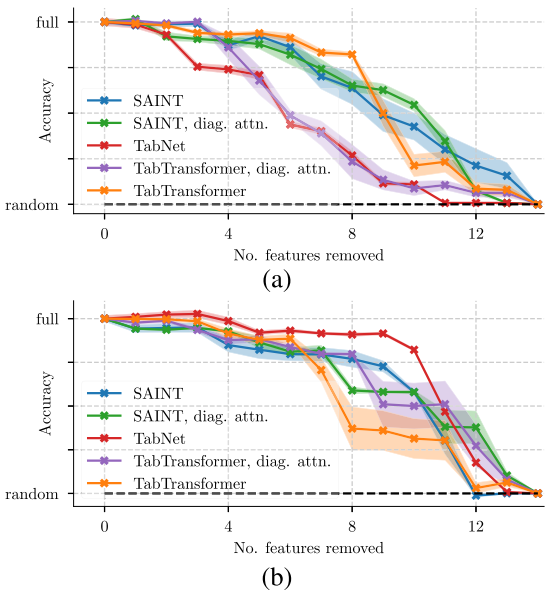

Fig. 6. Resulting curves of the global attribution benchmark for feature attributions (fifteen runs on Adult). Standard errors are indicated by the shaded area. For the MoRF order, an early drop in accuracy is desirable, while for LeRF, the accuracy should stay as high as possible. (a) MoRF. (b) LeRF.

estimate or only take the diagonal (feature self-attentions) as attributions.

The obtained curves are visualized in Fig. 6. For the MoRF order, TabNet and TabTransformer with the diagonal of the attention head as attributions seem to perform best. For LeRF, TabNet is the only significantly better method than the others. For TabTransformer, taking the diagonal of the attention matrix seems to increase the performance, whereas for SAINT, there is almost no difference. We additionally compare the attribution values obtained to values from the KernelSHAP attribution method. Unfortunately, there are no ground-truth attributions to compare with. However, the SHAP framework has a solid grounding in game theory and is widely deployed [43]. We only compare the absolute values of the attributions, as the attention maps are constrained to be positive. As a measure of agreement, we compute the Spearman rank correlation between the attributions by the SHAP framework and the tabular data models and show the results in Table VI. The correlation we observe is surprisingly low across all models, and sometimes, it is even negative, which means that a higher SHAP attribution will probably result in a lower attribution by the model.

In these two simple benchmarks, the transformer models were not able to produce convincing feature attributions out-of-the-box. We come to the conclusion that more profound benchmarks of the claimed interpretability characteristics and their usefulness in practice are necessary.

VIII. Discussion and Future Prospects

In this section, we summarize our findings and discuss current and future trends in deep learning approaches for tabular data (Section VIII-A). Moreover, we identify several open research questions that could be tackled to advance the field of tabular deep neural networks (Section VIII-B).

TABLE VI
Spearman Rank Correlation of the Provided Attribution With KernelSHAP Values as Ground Truth. Results Were Computed on 750 Random Samples From the Adult Dataset

Model, attention used	Spearman Corr.
TabTransformer, columnw. attention	-0.01 ± 0.008
TabTransformer, diag. attention	0.00 ± 0.010
TabNet	**0.07** ± 0.009
SAINT, columnw. attention	-0.04 ± 0.007
SAINT, diag. attention	0.01 ± 0.007

A. Summary and Trends

1) Decision Tree Ensembles Are Still State of the Art: In a fair comparison on multiple datasets, we demonstrated that models based on tree ensembles, such as XGBoost, LightGBM, and CatBoost, still outperform the deep learning models on most datasets that we considered and come with the additional advantage of significantly less training time. Even though it has been six years since the XGBoost publication [46] and over 20 years since the publishing of original gradient boosting paper [95], we can state that despite much research effort in deep learning, the state of the art for tabular data remains largely unchanged. However, we observed that for very large datasets, approaches based on deep learning may still be able to achieve competitive performance and even outperform classical models. In summary, we think that a fundamental reorientation of the domain may be necessary. For now, the question of whether the use of current deep learning techniques is beneficial for tabular data can generally be answered in the negative. This applies in particular to small heterogeneous datasets that are common in applications. Hence, instead of proposing more and more complex models, we argue that a more profound understanding of the reasons for this performance gap is needed.

2) Unified Benchmarking: Furthermore, our results highlight the need for unified benchmarks. There is no consensus in the machine learning community on how to make a fair and efficient comparison. Shwartz-Ziv and Armon [8] showed that the choice of benchmarking datasets can have a non-negligible impact on the performance assessment. While we chose common datasets with varying characteristics for our experiments, a different choice of datasets or hyperparameter such as the encoding use (e.g., one-hot encoding for categorical variables) may lead to a different outcome. Because of the excessive number of datasets (in the 18 works listed in Table II, over 100 different datasets are used), there is a necessity for a standardized benchmarking procedure, which allows to identify significant progress with respect to the state of the art. With this work, we also propose an open-source benchmark for deep learning models on tabular data. For tabular data generation tasks, Xu et al. [130] proposed a sound evaluation framework with artificial and real-world datasets (Section V-B), but researchers need to agree on common benchmarks in this subdomain as well.

3) Tabular Data Preprocessing: Many of the challenges for deep neural networks on tabular data are related to the heterogeneity of the data (e.g., categorical and sparse values).

Therefore, some deep learning solutions transform them into a homogeneous representation more suitable to neural networks. While the additional overhead is small, such transforms can boost performance considerably and should thus be among the first strategies applied in real-world scenarios.

4) Architectures for Deep Learning on Tabular Data: Architecturewise, there has been a clear trend toward transformer-based solutions (Section IV-B2) in recent years. These approaches offer multiple advantages over standard neural network architectures, for instance, learning with attention over both categorical and numerical features. Moreover, self-supervised or unsupervised pretraining that leverages unlabeled tabular data to train parts of the deep learning model is gaining popularity, not only among transformer-based approaches. Performancewise, multiple independent evaluations demonstrate that deep neural network methods from the hybrid (Section IV-B1) and transformer-based (Section IV-B2) groups exhibit superior predictive performance compared to plain deep neural networks on various datasets [9], [48], [62], [84]. This underlines the importance of special-purpose architectures for tabular data.

5) Deep Generative Models for Tabular Data: Powerful tabular data generation is essential for the development of high-quality models, particularly in a privacy context. With suitable data generators at hand, developers can use large, synthetic, and yet realistic datasets to develop better models, while not being subject to privacy concerns [145]. Unfortunately, the generation task is as hard as inference in predictive models, so progress in both areas will likely go hand in hand.

6) Interpretable Deep Learning Models for Tabular Data: Interpretability is undoubtedly desirable, particularly for tabular data models frequently applied to personal data, e.g., in healthcare and finance. An increasing number of approaches offer it out-of-the-box, but most current deep neural network models are still mainly concerned with the optimization of a chosen error metric. Therefore, extending existing open-source libraries (see [157], [170]) aimed at interpreting black-box models helps advance the field. Moreover, interpretable deep tabular learning is essential for understanding model decisions and results, especially for life-critical applications. However, much of the state-of-the-art recourse literature does not offer easy support of heterogeneous tabular data and lacks metrics to evaluate the quality of heterogeneous data recourse. Finally, model explanations can be used to identify and mitigate potential unwanted biases and eliminate unfair discrimination [204], [205].

7) Learning From Evolving Data Streams: Many modern applications are subject to continuously evolving data streams, e.g., social media, online retail, or healthcare. Streaming data are usually heterogeneous and potentially unlimited. Therefore, observations must be processed in a single pass and cannot be stored. Indeed, online learning models can only access a fraction of the data at each time step. Furthermore, they have to deal with limited resources and shifting data distributions (i.e., concept drift) [206]. Hence, hyperparameter optimization and model selection, as typically involved in deep learning, are usually not feasible in a data stream. For this reason, despite the success of deep learning in other domains, less complex methods, such as incremental decision trees [207], [208], are often preferred in online learning applications.

B. Open Research Questions

Several open problems need to be addressed in future research. In this section, we will list those we deem fundamental to the domain.

1) Information-Theoretic Analysis of Encodings: Encoding methods are highly popular when dealing with tabular data. However, the majority of data preprocessing approaches for deep neural networks are lossy in terms of information content. Therefore, it is challenging to achieve an efficient, almost lossless transformation of heterogeneous tabular data into homogeneous data. Nevertheless, the information-theoretic view on these transformations remains to be investigated in detail and could shed light on the underlying mechanisms.

2) Computational Efficiency in Hybrid Models: The work by Shwartz-Ziv and Armon [8] suggests that the combination of a GBDT and deep neural networks may improve the predictive performance of a machine learning system. However, it also leads to growing complexity. Training or inference times, which far exceed those of classical machine learning approaches, are a recurring problem when developing hybrid models. We conclude that the integration of state-of-the-art approaches from classical machine learning and deep learning has not been conclusively resolved yet and future work should be conducted on how to mitigate the tradeoff between predictive performance and computational complexity.

3) Individual Regularizations: We applaud recent research on individual regularization methods, in which we see a promising direction to tackle the problem of highly sensitive features. We believe that representing the towering influence of certain features is crucial to success. Whether context- and architecture-specific regularizations for tabular data can be found remains an open question. In addition, it is relevant to explore the theoretical constraints that govern the success of regularization on tabular data more profoundly.

4) Novel Processes for Tabular Data Generation: For tabular data generation, modified GANs and VAEs are prevalent. However, the modeling of dependencies and categorical distributions remains the key challenge. Novel architectures in this area, such as diffusion models, have not been adapted to the domain of tabular data. Furthermore, the definition of an entirely new generative process particularly focused on tabular data might be worth investigating.

5) Interpretability: Going forward, counterfactual explanations for deep tabular learning can be used to improve the perceived fairness in human–artificial intelligence (AI) interaction scenarios and to enable personalized decision-making [188]. However, the heterogeneity of tabular data poses problems for counterfactual explanation methods to be reliably deployed in practice. The problem of efficiently handling heterogeneous tabular data in the presence of feasibility constraints remains unsolved [157].

6) Transfer of Deep Learning Methods to Data Streams: Recent work shows that some of the limitations of neural networks in an evolving data stream can be overcome [25], [209]. Conversely, changes in the parameters of a neural

network may be effectively used to weigh the importance of input features over time [210] or to detect concept drift [211]. Accordingly, we argue that deep learning for streaming data—in particular strategies for dealing with evolving and heterogeneous tabular data—should receive more attention in the future.

7) Transfer Learning for Tabular Data: Reusing knowledge gained solving one problem and applying it to a different task is the research problem addressed by transfer learning. While transfer learning is successfully used in computer vision and natural language processing applications [212], there are no efficient and generally accepted ways to do transfer learning for tabular data. Hence, a general research question can be how to share knowledge between multiple (related) tabular datasets efficiently.

8) Data Augmentation for Tabular Data: Data augmentation has proven highly effective to prevent overfitting, especially in computer vision [213]. While some data augmentation techniques for tabular data exist, e.g., SMOTE-NC [214], simple models fail to capture the dependency structure of the data. Therefore, generating additional samples in a continuous latent space is a promising direction. This was investigated by Darabi and Elor [37] for minority oversampling. Nevertheless, the reported improvements are only marginal. Thus, future work is required to find simple, yet effective random transformations to enhance tabular training sets.

9) Self-Supervised Learning: Large-scale labeled data are usually required to train deep neural networks; however, data labeling is an expensive task. To avoid this expensive step, self-supervised methods propose to learn general feature representations from available unlabeled data. These methods have also shown astonishing results in computer vision and natural language processing [215], [216]. Only a few recent works in this direction [79], [80], [217] deal with heterogeneous data. Hence, novel self-supervised learning approaches dedicated to tabular data might be worth investigating.

IX. Conclusion

This survey is the first work to systematically explore deep neural network approaches for heterogeneous tabular data. In this context, we highlighted the main challenges and research advances in modeling, generating, and explaining tabular data. We introduced a unified taxonomy that categorizes deep learning approaches for tabular data into three branches: data transformation methods, specialized architectures, and regularization models. We believe that our taxonomy will help catalog future research and better understand and address the remaining challenges in applying deep learning to tabular data. We hope that it will help researchers and practitioners to find the most appropriate strategies and methods for their applications.

In addition, we also conducted an unbiased evaluation of the state-of-the-art deep learning approaches on multiple real-world datasets. Deep neural network-based methods for heterogeneous tabular data are still inferior to machine learning methods based on decision tree ensembles for small- and medium-sized datasets (less than $\sim$1M samples). Only for a very large dataset mainly consisting of continuous and numerical variables, the deep learning model SAINT outperformed these classical approaches. Furthermore, we assessed the explanation properties of deep learning models with the self-attention mechanism. Although the TabNet model shows promising explanatory capabilities, inconsistencies between the explanations remain an open issue.

Due to the importance of tabular data to industry and academia, new ideas in this area are in high demand and can have a significant impact. With this review, we hope to provide interested readers with the references and insights they need to address open challenges and effectively advance the field.

References

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, May 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[5] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[6] S. O. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," 2019, *arXiv:1908.07442*.

[7] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," 2019, *arXiv:1909.06312*.

[8] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," 2021, *arXiv:2106.03253*.

[9] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training," 2021, *arXiv:2106.01342*.

[10] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka, "Well-tuned simple nets excel on tabular datasets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–14.

[11] D. Ulmer, L. Meijerink, and G. Cinà, "Trust Issues: Uncertainty estimation does not enable reliable OOD detection on medical tabular data," in *Proc. Mach. Learn. Health NeurIPS Workshop*, 2020, pp. 341–354.

[12] S. Somani et al., "Deep learning and the electrocardiogram: Review of the current state-of-the-art," *EP Europace*, vol. 23, no. 8, pp. 1179–1191, Aug. 2021.

[13] V. Borisov, E. Kasneci, and G. Kasneci, "Robust cognitive load detection from wrist-band sensors," *Comput. Hum. Behav. Rep.*, vol. 4, Aug. 2021, Art. no. 100116.

[14] J. M. Clements, D. Xu, N. Yousefi, and D. Efimov, "Sequential deep learning for credit risk monitoring with tabular financial data," 2020, *arXiv:2012.15330*.

[15] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*.

[16] Z. Shuai, L. Yao, A. Sun, and T. Yi, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2017.

[17] Q. Zhang, L. Cao, C. Shi, and Z. Niu, "Neural time-aware sequential recommendation by jointly modeling preference dynamics and explicit feature couplings," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5125–5137, Oct. 2022.

[18] M. Ahmed, H. Afzal, A. Majeed, and B. Khan, "A survey of evolution in predictive models and impacting factors in customer churn," *Adv. Data Sci. Adapt. Anal.*, vol. 9, no. 3, Jul. 2017, Art. no. 1750007.

[19] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[20] F. Cartella, O. Anunciação, Y. Funabiki, D. Yamaguchi, T. Akishita, and O. Elshocht, "Adversarial attacks for tabular data: Application to fraud detection and imbalanced data," in *Proc. CEUR Workshop*, vol. 2808, 2021, pp. 1–9.

[21] C. J. Urban and K. M. Gates, "Deep learning: A primer for psychologists," *Psychol. Methods*, vol. 26, no. 6, pp. 743–773, 2021.

[22] G. Pang, C. Aggarwal, C. Shen, and N. Sebe, "Editorial deep learning for anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2282–2286, Jun. 2022.

[23] S. Wang et al., "Multiview deep anomaly detection: A systematic exploration," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 26, 2022, doi: 10.1109/TNNLS.2022.3184723.

[24] V. Škvára, J. Francå, M. Zorek, T. Pevnỳ, and V. Šmídl, "Comparison of anomaly detectors: Context matters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2494–2507, Jun. 2022.

[25] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online deep learning: Learning deep neural networks on the fly," 2017, *arXiv:1711.03705*.

[26] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106622.

[27] P. Yin, G. Neubig, W.-T. Yih, and S. Riedel, "TaBERT: Pretraining for joint understanding of textual and tabular data," 2020, *arXiv:2005.08314*.

[28] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," 2019, *arXiv:1906.01529*.

[29] D. Lichtenwalter, P. Burggräf, J. Wagner, and T. Weißer, "Deep multimodal learning for manufacturing problem solving," *Proc. CIRP*, vol. 99, pp. 615–620, 2021.

[30] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.

[31] D. Medvedev and A. D'yakonov, "New properties of the data distillation method when working with tabular data," 2020, *arXiv:2010.09839*.

[32] J. Li, Y. Li, X. Xiang, S.-T. Xia, S. Dong, and Y. Cai, "TNT: An interpretable tree-network-tree learning framework using knowledge distillation," *Entropy*, vol. 22, no. 11, p. 1203, Oct. 2020.

[33] D. Roschewitz, M.-A. Hartley, L. Corinzia, and M. Jaggi, "IFedAvg: Interpretable data-interoperability for federated learning," 2021, *arXiv:2107.06580*.

[34] A. Sánchez-Morales, J.-L. Sancho-Gómez, J.-A. Martínez-García, and A. R. Figueiras-Vidal, "Improving deep learning performance with missing values via deletion and compensation," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13233–13244, Sep. 2020.

[35] M. Abroshan, K. H. Yip, C. Tekin, and M. Van Der Schaar, "Conservative policy construction using variational autoencoders for logged data with missing values," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 10, 2022, doi: 10.1109/TNNLS.2021.3136385.

[36] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114582.

[37] S. Darabi and Y. Elor, "Synthesising multi-modal minority samples for tabular data," 2021, *arXiv:2105.08204*.

[38] S. Kamthe, S. Assefa, and M. Deisenroth, "Copula flows for synthetic data generation," 2021, *arXiv:2101.00598*.

[39] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Proc. 2nd Mach. Learn. Healthcare Conf.*, 2017, pp. 286–305.

[40] State of California, Department of Justice. (2018). *California Consumer Privacy Act (CCPA)*. Accessed: Dec. 20, 2022. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[41] GDPR. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council*. Official Journal of the European Union. [Online]. Available: http://www.privacyregulation.eu/en/13.htm

[42] P. Voigt and A. Von Dem Bussche, "The EU general data protection regulation (GDPR)," in *A Practical Guide*, vol. 10, 1st ed. Cham, Switzerland: Springer, 2017, Art. no. 3152676.

[43] M. Sahakyan, Z. Aung, and T. Rahwan, "Explainable artificial intelligence for tabular data: A survey," *IEEE Access*, vol. 9, pp. 135392–135422, 2021.

[44] B. I. Grisci, M. J. Krause, and M. Dorn, "Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data," *Inf. Sci.*, vol. 559, pp. 111–129, Jun. 2021.

[45] U. Bhatt et al., "Explainable machine learning in deployment," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 648–657.

[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[47] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J. Big Data*, vol. 7, no. 1, pp. 1–41, Dec. 2020.

[48] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," 2021, *arXiv:2106.11959*.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[50] L. Katzir, G. Elidan, and R. El-Yaniv, "Net-DNF: Effective deep modeling of tabular data," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.

[51] R. U. David and M. Lane, *Introduction to Statistics*. 2003. [Online]. Available: http://onlinestatbook.com/

[52] M. Ryan, *Deep Learning With Structured Data*. New York, NY, USA: Simon & Schuster, 2020.

[53] M. W. Cvitkovic et al., "Deep learning in unconventional domains," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 2020.

[54] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[55] A. J. Miles, "The sunstroke epidemic of Cincinnati, Ohio, during the summer of 1881," *Public Health Papers Rep.*, vol. 7, no. 1, pp. 293–304, 1881.

[56] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Aug. 1936.

[57] D. A. Jdanov, D. Jasilionis, V. M. Shkolnikov, and M. Barbieri, "Human mortality database," in *Encyclopedia Gerontology Population Aging*, D. Gu and M. E. Dupre, Eds. Cham, Switzerland: Springer, 2020.

[58] E. Fix, *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties*. Wright-Patterson AFB, OH, USA: USAF school of Aviation Medicine, 1951.

[59] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, and Y. C. Lee, "Learning and extracting finite state automata with second-order recurrent neural networks," *Neural Comput.*, vol. 4, no. 3, pp. 393–405, May 1992.

[60] L. Willenborg and T. De Waal, *Statistical Disclosure Control in Practice*, vol. 111. New York, NY, USA: Springer, 1996.

[61] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 521–530.

[62] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu, "DeepGBM: A deep learning framework distilled by GBDT for online prediction tasks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 384–394.

[63] I. Shavitt and E. Segal, "Regularization learning networks: Deep learning for tabular datasets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1379–1389.

[64] T. B. Brown et al., "Language models are few-shot learners," 2020, *arXiv:2005.14165*.

[65] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–11.

[66] S. Khan, M. Naseer, M. Hayat, S. Waqas Zamir, F. Shahbaz Khan, and M. Shah, "Transformers in vision: A survey," 2021, *arXiv:2101.01169*.

[67] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021.

[68] A. F. Karr, A. P. Sanil, and D. L. Banks, "Data quality: A statistical perspective," *Stat. Methodol.*, vol. 3, no. 2, pp. 137–173, 2006.

[69] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," 2018, *arXiv:1811.11264*.

[70] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.

[71] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.

[72] Y. Zhu et al., "Converting tabular data into images for deep learning with convolutional neural networks," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, May 2021.

[73] N. Rahaman et al., "On the spectral bias of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5301–5310.

[74] B. R. Mitchell et al., "The spatial inductive bias of deep learning," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, USA, 2017.

[75] Y. Gorishniy, I. Rubachev, and A. Babenko, "On embeddings for numerical features in tabular deep learning," 2022, *arXiv:2203.05556*.

[76] E. Fitkov-Norris, S. Vahid, and C. Hand, "Evaluating the impact of categorical data encoding and scaling on neural network classification performance: The case of repeat consumption of identical cultural goods," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Cham, Switzerland: Springer, 2012, pp. 343–352.

[77] D. Baylor et al., "TFX: A TensorFlow-based production-scale machine learning platform," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1387–1395.

[78] B. Sun et al., "SuperTML: Two-dimensional word embedding for the precognition on structured tabular data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9.

[79] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, "VIME: Extending the success of self- and semi-supervised learning to tabular domaindim," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–11.

[80] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "SCARF: Self-supervised contrastive learning using random feature corruption," 2021, *arXiv:2106.15147*.

[81] H.-T. Cheng et al., "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[82] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," 2017, *arXiv:1711.09784*.

[83] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1754–1763.

[84] G. Ke, J. Zhang, Z. Xu, J. Bian, and T.-Y. Liu. (2018). *TabNN: A Universal Neural Network Solution for Tabular Data*. [Online]. Available: https://openreview.net/forum?id=r1eJssCqY7

[85] R. Agarwal et al., "Neural additive models: Interpretable machine learning with neural nets," 2020, *arXiv:2004.13912*.

[86] Y. Luo, H. Zhou, W.-W. Tu, Y. Chen, W. Dai, and Q. Yang, "Network on network for tabular data classification in real-world applications," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 2317–2326.

[87] Z. Liu, Q. Liu, H. Zhang, and Y. Chen, "DNN2LR: Interpretation-inspired feature crossing for real-world tabular data," 2020, *arXiv:2008.09775*.

[88] S. Ivanov and L. Prokhorenkova, "Boost then Convolve: Gradient boosting meets graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.

[89] H. Luo, F. Cheng, H. Yu, and Y. Yi, "SDTR: Soft decision tree regressor for tabular data," *IEEE Access*, vol. 9, pp. 55999–56011, 2021.

[90] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular data modeling using contextual embeddings," 2020, *arXiv:2012.06678*.

[91] S. Cai, K. Zheng, G. Chen, H. V. Jagadish, B. C. Ooi, and M. Zhang, "ARM-Net: Adaptive relation modeling network for structured data," in *Proc. Int. Conf. Manage. Data*, Jun. 2021, pp. 207–220.

[92] J. Kossen, N. Band, C. Lyle, A. Gomez, T. Rainforth, and Y. Gal, "Self-attention between datapoints: Going beyond individual input-output pairs in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 28742–28756.

[93] Y. Yamada, O. Lindenbaum, S. Negahban, and Y. Kluger, "Feature selection using stochastic gates," in *Proc. Mach. Learn. Syst.*, 2020, pp. 8952–8963.

[94] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *ACM SIGKDD Explor. Newslett.*, vol. 3, no. 1, pp. 27–32, Jul. 2001.

[95] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002.

[96] P. Langley and S. Sage, "Oblivious decision trees and abstract cases," in *Proc. Work. Notes AAAI Workshop Case-Based Reasoning*. Seattle, WA, USA, 1994, pp. 113–117.

[97] B. Peters, V. Niculae, and A. F. T. Martins, "Sparse Sequence-to-Sequence models," 2019, *arXiv:1905.05702*.

[98] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[99] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," 2016, *arXiv:1604.06737*.

[100] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.

[101] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Proc. 20th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2006, pp. 985–992.

[102] X. He et al., "Practical lessons from predicting clicks on ads at Facebook," in *Proc. 8th Int. Workshop Data Mining Online Advertising*, 2014, pp. 1–9.

[103] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[104] C. Wang, M. Li, and A. J. Smola, "Language models with transformers," 2019, *arXiv:1904.09408*.

[105] A. F. T. Martins and R. Fernandez Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," 2016, *arXiv:1602.02068*.

[106] G. Van Rossum and F. L. Drake, Jr., *Python Reference Manual*. Amsterdam, The Netherlands: Centrum voor Wiskunde en Informatica, 1995.

[107] M. Joseph, "PyTorch tabular: A framework for deep learning with tabular data," 2021, *arXiv:2104.13638*.

[108] S. Boughorbel, F. Jarray, and A. Kadri, "Fairness in TabNet model by disentangled representation for the prediction of hospital no-show," 2021, *arXiv:2103.04048*.

[109] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, Jul. 2021.

[110] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.

[111] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[112] V. Borisov, J. Haug, and G. Kasneci, "CancelOut: A layer for feature selection in deep neural networks," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 72–83.

[113] G. Valdes, W. Arbelo, Y. Interian, and J. H. Friedman, "Lockout: Sparse regularization of neural networks," 2021, *arXiv:2107.07160*.

[114] J. Fiedler, "Simple modifications to improve tabular neural networks," 2021, *arXiv:2108.03214*.

[115] K. Lounici, K. Meziani, and B. Riu, "Muddling label regularization: Deep learning for tabular datasets," 2021, *arXiv:2106.04462*.

[116] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 1–10.

[117] H. Chen, S. Jajodia, J. Liu, N. Park, V. Sokolov, and V. S. Subrahmanian, "FakeTables: Using GANs to generate functional dependency preserving tables with bounded real data," in *Proc. Twenty-Eighth Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2074–2080.

[118] L. Gondara and K. Wang, "MIDA: Multiple imputation using denoising autoencoders," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2018, pp. 260–272.

[119] R. D. Camino et al., "Working with deep generative models and tabular data imputation," in *Proc. ICML Artemiss Workshop*, 2020, pp. 1–6.

[120] M. Quintana and C. Miller, "Towards class-balancing human comfort datasets with GANs," in *Proc. 6th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2019, pp. 391–392.

[121] A. Koivu, M. Sairanen, A. Airola, and T. Pahikkala, "Synthetic minority oversampling of vital statistics data with generative adversarial networks," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 11, pp. 1667–1674, Nov. 2020.

[122] J. Fan, J. Chen, T. Liu, Y. Shen, G. Li, and X. Du, "Relational data synthesis using generative adversarial networks: A design space exploration," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 1962–1975, Aug. 2020.

[123] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[124] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[125] S. Subramanian, S. Rajeswar, F. Dutil, C. Pal, and A. Courville, "Adversarial generation of natural language," in *Proc. 2nd Workshop Represent. Learn. NLP*, 2017, pp. 241–251.

[126] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 399–410.

[127] Z. Li, Y. Zhao, and J. Fu, "SynC: A copula based framework for generating synthetic data from aggregated sources," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2020, pp. 571–578.

[128] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," *ACM Trans. Database Syst.*, vol. 42, no. 4, pp. 1–41, Oct. 2017.

[129] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 462–467, May 1968.

[130] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2019, pp. 1–11.

[131] L. V. H. Vardhan and S. Kok, "Generating privacy-preserving synthetic tabular data using oblivious variational autoencoders," in *Proc. Workshop Econ. Privacy Data Labor 37th Int. Conf. Mach. Learn.*, 2020, pp. 1–8.

[132] M. Baak, S. Brugman, I. F. Rojas, L. Dalmeida, R. E. Urlus, and J.-B. Oger, "Synthsonic: Fast, probabilistic modeling and synthesis of tabular data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 4747–4763.

[133] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[134] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–16.

[135] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[136] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[137] M. G. Bellemare et al., "The Cramér distance as a solution to biased Wasserstein gradients," 2017, *arXiv:1705.10743*.

[138] R. D. Hjelm, A. P. Jacob, T. Che, A. Trischler, K. Cho, and Y. Bengio, "Boundary-seeking generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[139] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3310–3320.

[140] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR), Conf. Track*, 2014, pp. 1–14.

[141] C. Ma, S. Tschiatschek, R. Turner, J. M. Hernández-Lobato, and C. Zhang, "VAEM: A deep generative model for heterogeneous mixed type data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–11.

[142] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, Jun. 2018.

[143] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.

[144] B. Wen, L. O. Colon, K. P. Subbalakshmi, and R. Chandramouli, "Causal-TGAN: Generating tabular data using causal generative adversarial networks," 2021, *arXiv:2104.10680*.

[145] J. Jordon, J. Yoon, and M. Van Der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–21.

[146] N. M. Jebreel, J. Domingo-Ferrer, A. Blanco-Justicia, and D. Sánchez, "Enhanced security and privacy via fragmented federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 19, 2022, doi: 10.1109/TNNLS.2022.3212627.

[147] A. Mottini, A. Lheritier, and R. Acuna-Agost, "Airline passenger name record generation using generative adversarial networks," 2018, *arXiv:1807.06657*.

[148] R. Camino, C. Hammerschmidt, and R. State, "Generating multi-categorical samples with generative adversarial networks," in *Proc. ICML Workshop Theor. Found. Appl. Deep Generative Models*, 2018, pp. 1–7.

[149] M. K. Baowaly, C.-C. Lin, C.-L. Liu, and K.-T. Chen, "Synthesizing electronic health records using improved generative adversarial networks," *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 3, pp. 228–241, Mar. 2019.

[150] Z. Zhao, A. Kunar, H. Van der Scheer, R. Birke, and L. Y. Chen, "CTAB-GAN: Effective table data synthesizing," 2021, *arXiv:2102.08369*.

[151] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, and G. Kasneci, "Language models are realistic tabular data generators," 2022, *arXiv:2210.06280*.

[152] F. J. Massey, Jr., "The Kolmogorov-Smirnov test for goodness of fit," *J. Amer. Statist. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[153] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021.

[154] J. Kauffmann, M. Esders, L. Ruff, G. Montavon, W. Samek, and K.-R. Müller, "From clustering to cluster explanations via neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 7, 2022, doi: 10.1109/TNNLS.2022.3185901.

[155] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Sep. 2019.

[156] K. Gade, S. C. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, "Explainable AI in industry," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3203–3204.

[157] M. Pawelczyk, S. Bielawski, J. Van Den Heuvel, T. Richter, and G. Kasneci, "CARLA: A Python library to benchmark algorithmic recourse and counterfactual explanation algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Benchmark Datasets Track*, 2021, pp. 1–22.

[158] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 150–158.

[159] D. Alvarez-Melis and T. S. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *Proc. NeurIPS*, 2018, pp. 1–10.

[160] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing theory-driven user-centric explainable AI," in *Proc. CHI*, 2019, pp. 1–15.

[161] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[162] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham, Switzerland: Springer, 2019, pp. 193–209.

[163] G. Kasneci and T. Gottron, "LICON: A linear weighting scheme for the contribution ofInput variables in deep artificial neural networks," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 45–54.

[164] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[165] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1–9.

[166] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[167] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI*, 2018, pp. 1–9.

[168] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," *Nature Mach. Intell.*, vol. 2, pp. 56–67, Jan. 2020.

[169] J. Haug, S. Zürn, P. El-Jiz, and G. Kasneci, "On baselines for local feature attributions," 2021, *arXiv:2101.00905*.

[170] Y. Liu, S. Khandagale, C. White, and W. Neiswanger, "Synthetic benchmarks for scientific research in explainable machine learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS) Benchmark Datasets Track*, 2021, pp. 1–25.

[171] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, p. 841, 2018.

[172] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 10–19.

[173] C. Russell, "Efficient search for diverse coherent explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 20–28.

[174] K. Rawal and H. Lakkaraju, "Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses," in *Proc. NeurIPS*, 2020, pp. 12187–12198.

[175] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Model-agnostic counterfactual explanations for consequential decisions," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 895–905.

[176] A. Dhurandhar et al., "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.

[177] B. Mittelstadt, C. Russell, and S. Wachter, "Explaining explanations in AI," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 279–288.
[178] M. Pawelczyk, K. Broelemann, and G. Kasneci, "Learning model-agnostic counterfactual explanations for tabular data," in *Proc. Web Conf.*, Apr. 2020, pp. 3126–3132.
[179] M. Downs, J. L. Chu, Y. Yacoby, F. Doshi-Velez, and W. Pan, "CRUDS: Counterfactual recourse using disentangled subspaces," in *Proc. ICML Workshop Hum. Interpretability Mach. Learn. (WHI)*, 2020, 1–23.
[180] S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh, "Towards realistic individual recourse and actionable explanations in black-box decision making systems," 2019, *arXiv:1907.09615*.
[181] D. Mahajan, C. Tan, and A. Sharma, "Preserving causal constraints in counterfactual explanations for machine learning classifiers," 2019, *arXiv:1912.03277*.
[182] M. Pawelczyk, K. Broelemann, and G. Kasneci, "On counterfactual explanations under predictive multiplicity," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, 2020, pp. 809–818.
[183] A. Nazábal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using VAEs," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107501.
[184] S. Upadhyay, S. Joshi, and H. Lakkaraju, "Towards robust and reliable algorithmic recourse," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp.16926–16937.
[185] R. Dominguez-Olmedo, A.-H. Karimi, and B. Schölkopf, "On the adversarial robustness of causal algorithmic recourse," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 5324–5342.
[186] M. Pawelczyk, T. Datta, J. Van-Den-Heuvel, G. Kasneci, and H. Lakkaraju, "Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse," 2022, *arXiv:2203.06768*.
[187] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera, "A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects," 2020, *arXiv:2010.04050*.
[188] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," 2020, *arXiv:2010.10596*.
[189] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[190] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[191] FICO. (2019). *Home Equity Line of Credit (HELOC) Dataset*. Accessed: Jun. 15, 2022. [Online]. Available: https://community.fico.com/s/explainable-machine-learning-challenge
[192] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Commun.*, vol. 5, no. 1, pp. 1–9, Sep. 2014.
[193] C. Z. Mooney, *Monte Carlo Simulation*. Newbury Park, CA, USA: SAGE, 1997.
[194] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statist. Probab. Lett.*, vol. 33, pp. 291–297, May 1997.
[195] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification Regression Trees*. Evanston, IL, USA: Routledge, 2017.
[196] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[197] K. Broelemann and G. Kasneci, "A gradient-based split criterion for highly accurate and transparent model trees," in *Proc. IJCAI*, 2019, pp. 1–8.
[198] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
[199] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1–10.
[200] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, p. 2, 2014.
[201] C. S. Bojer and J. P. Meldgaard, "Kaggle forecasting competitions: An overlooked learning opportunity," *Int. J. Forecasting*, vol. 37, no. 2, pp. 587–603, Apr. 2021.
[202] Y. Rong, T. Leemann, V. Borisov, G. Kasneci, and E. Kasneci, "A consistent and efficient evaluation strategy for attribution methods," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18770–18795.
[203] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity checks for saliency metrics," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 6021–6029.
[204] E. Ntoutsi et al., "Bias in data-driven artificial intelligence systems-an introductory survey," *Wiley Interdiscipl. Reviews: Data Mining Knowl. Discovery*, vol. 10, no. 3, p. e1356, 2020.
[205] A. Giloni et al., "BENN: Bias estimation using a deep neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 11, 2022, doi: 10.1109/TNNLS.2022.3172365.
[206] Y. Sun, K. Tang, Z. Zhu, and X. Yao, "Concept drift adaptation by exploiting historical knowledge," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4822–4832, Oct. 2018.
[207] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2000, pp. 71–80.
[208] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1953–1962.
[209] P. Duda, M. Jaworski, A. Cader, and L. Wang, "On training deep neural networks using a streaming approach," *J. Artif. Intell. Soft Comput. Res.*, vol. 10, no. 1, pp. 15–26, Jan. 2020.
[210] J. Haug, M. Pawelczyk, K. Broelemann, and G. Kasneci, "Leveraging model inherent variable importance for stable online feature selection," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1478–1502.
[211] J. Haug and G. Kasneci, "Learning parameter distributions to detect concept drift in data streams," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 9452–9459.
[212] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 270–279.
[213] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[214] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, Jan. 2002.
[215] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.
[216] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2021.
[217] T. Ucar, E. Hajiramezanali, and L. Edwards, "SubTab: Subsetting features of tabular data for self-supervised representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–13.