Navigating Tabular Data Synthesis Research

Understanding User Needs and Tool Capabilities

Maria F. Davila R.[1, 4], Sven Groen[2], Fabian Panse[3], Wolfram Wingerath[4]
[1]OFFIS - Institute for Informatics, Germany
[2]Initions Consulting GmbH, Germany
[3]Hasso Plattner Institute, Germany
[4]University of Oldenburg, Germany
maria.davila@offis.de, sven.groen@initions-consulting.com
fabian.panse@hpi.de, wolfram.wingerath@uol.de

ABSTRACT

In an era of rapidly advancing data-driven applications, there is a growing demand for data in both research and practice. Synthetic data have emerged as an alternative when there are not enough real data available or when these data may not be shared (e.g., due to privacy regulations). Synthesizing tabular data presents unique and complex challenges, especially handling (i) missing values, (ii) dataset imbalance, (iii) diverse column types, and (iv) complex data distributions, as well as preserving (v) column correlations, (vi) temporal dependencies, and (vii) integrity constraints (e.g., functional dependencies) present in the original dataset. Although significant progress has been made recently in the development of generational models, there is no one-size-fits-all solution for tabular data today and choosing the right tool for a particular use case remains a difficult task.

In this paper, we survey the state of the art in Tabular Data Synthesis (TDS) and examine user needs by defining a set of functional and non-functional requirements. We also evaluate the reported performance of 37 TDS research tools on these requirements, develop a tool selection guide to help users find a suitable TDS tool for their use case, and identify open challenges in TDS research, especially with respect to data management.

Keywords

Tabular data synthesis, Deep generative models, Functional requirements, Customized tool selection

1 Introduction

Nowadays, data are one of the most valuable resources but they are often not available in the required quantity or may not be shared between data consumers to protect data privacy or business interests. This hinders digital innovation and leaves a lot of potential untapped [84, 102]. The generation and sharing of synthetic data has emerged as a solution to this problem [27, 29, 43] and is therefore an important tool in many areas of data science, data engineering, and data management.

In the last few years, data generation has gained new momentum with the success of generative deep learning, particularly for generating synthetic images [23] and texts [89]. Although less visible in today's media culture, tabular data plays an essential role in many processing tasks. In this study, we consider tabular data according to the relational data model [20, 32], i.e., a tabular dataset consists of one or more tables, which in turn are organized into rows (or records) representing individual data points and columns representing different features of those data points.

Two branches of research have emerged for the generation of tabular data, which deal with different use cases: The first branch involves the controlled generation of new data based on schema information, statistics, and domain knowledge in order to benchmark newly developed database technologies such as hardware components, join operators, or query optimization strategies. Although the generated data should be realistic, the focus of these approaches is more on efficient, distributed, and scalable generation techniques to enable the creation of highly challenging benchmark datasets. Many of these approaches are rule-based and tailored to a specific domain [10, 18, 38, 45, 48, 81, 88].

The second branch focuses on replicating the characteristics of a given (real-world) dataset as accurately as possible. This is often combined with the additional requirement that no sensitive information from this dataset may be disclosed. This applies in particular to person-related information as we have it in medical [17, 43] or financial data [86]. The idea behind this is that the synthetic dataset can now be shared with others so that they can use it instead of the real dataset (e.g., for data repurposing or to outsource data-driven tasks), but the insights gained from their analyses also apply to the real dataset. A common area of application is machine learning, in which an ML model trained on synthetic data is applied to real data [62, 108].

In our study, we focus on the second branch of research which is usually called *tabular data synthesis* (TDS) [41, 55, 111, 116, 118]. Based on our research, there is currently no TDS tool that works well across all applications. Moreover, although commercial platforms, such as the Synthetic Data Vault (SDV) [85] or Gretel AI [7], choose and adapt TDS tools, there is currently no benchmark to measure a TDS tool's "fitness for use", which refers to the capability of the tool to meet the specific functional and non-functional requirements of a certain use case. Thus, selecting a TDS tool for a specific use case is a major challenge. This applies in particular to data management, which has additional functional requirements compared to other application areas such as machine learning. Especially important here is the ability to generate data with complex schemas (instead of individual tables) and to preserve integrity constraints, such as functional dependencies.

The main objective of our study is to provide the basis for evaluating the suitability of TDS tools for use-case specific requirements. Our contributions are:

c_1: a **survey** of current research on TDS,

c_2: the definition of **functional** and **non-functional requirements** that allow to assess a TDS tool's suitability for a specific use case,

c_3: an **assessment** of 37 tools with respect to their reported performance on those requirements,

c_4: a **tool selection guide** to help users find a suitable TDS tool for their use case, developed by compiling the reported performance of the leading TDS tools on such requirements, and

c_5: an overview of **research gaps** especially with respect to use cases in data management.

The remainder of the paper is structured as follows. Section 2 presents related work. In Section 3, we describe the main purposes for TDS and discuss potential use cases in data management. Section 4 describes the main challenges in TDS and its differences to image and text generation. These two sections are the basis for the identification of the functional and non-functional requirements in Section 5. While Section 6 provides background on TDS tools and their underlying models, Section 7 gives a brief overview of the evaluation of synthetic tabular data. The results of our study (i.e., assessment matrices, decision guide, and research gaps) are presented in Section 8. Finally, Section 9 concludes the paper and gives an outlook on our upcoming research.

2 Related Work

The Synthetic Data Vault (SDV) [85], Gretel AI [7], and Mostly AI [49] are platforms for the generation of tabular data. These platforms must choose from the available tools to address the widest range of use cases possible. However, these platforms do not report on the specific limitations of those tools. In contrast, our goal is to create a framework that allows to identify use-case specific requirements and determine those limitations.

Several surveys served as input to our work to identify the predominant TDS models and tools. Hernandez et al. [43], Fan et al. [27], Figueira et al, [29], and Brophy et al. [9] explored the use of Generative Adversarial Networks (GANs) for health records, categorical and numerical data types, and time series generation. Koo and Kim [60] reviewed generative diffusion models for tabular data, paralleling Lin et al. [68], who focused on time-series diffusion. Fonseca and Bacao [30] recently provide an extensive survey on tabular data synthesis including an evaluation of 70 tools across six different machine learning problems. However, while our focus is on various tools from the field of generative deep learning (including GANs, autoencoders, probabilistic diffusion, graph neural networks, and transformers), the only deep learning approaches they consider are GANs and autoencoders. Moreover, they do not address the problem of finding the most suitable tool for a specific use case and therefore do neither define functional and non-functional requirements for tabular data synthesis nor evaluate their tools in terms of those requirements.

In summary, none of these surveys provide a comparison of deep learning approaches (see Figure 1) as we do in this paper. Additionally, they do not provide any insights into how users can assess a tool's fitness for use, or guide them in the process of choosing a suitable TDS tool for their specific use case.

3 TDS Purposes and Use Cases

Based on our literature review, we pinpointed five purposes why users need to synthesize tabular data based on a given use-case specific dataset:

- **Privacy-Preserving Data Sharing:** Many domains contain sensitive data, requiring effective measures to protect privacy when these data need to be shared or reused. For instance, a hospital would like to share its electronic health records (EHRs) for external analyses and therefore creates synthetic patient records that closely resemble their real patient records but do not correspond to actual patient data.

- **Missing Value Imputation:** Datasets often have incomplete entries, which can distort analyses. In our EHR example, a patient's smoking status may be missing, which is vital information for predicting the risk of heart disease. TDS allows users to fill such gaps with meaningful values, ensuring intentional data completeness [79].

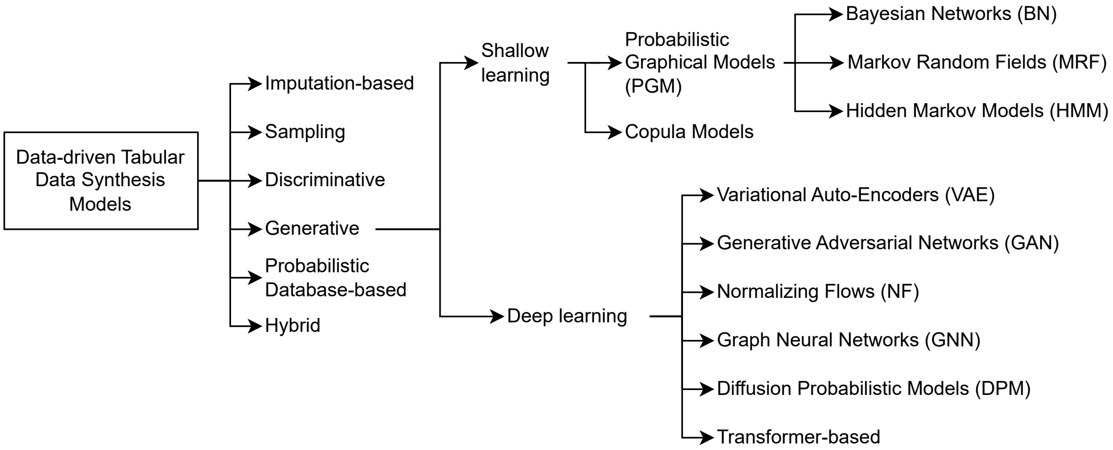
Data-driven Tabular Data Synthesis Models
Imputation-based
Sampling
Discriminative
Generative
Probabilistic Database-based
Hybrid
Shallow learning
Deep learning
Probabilistic Graphical Models (PGM)
Copula Models
Bayesian Networks (BN)
Markov Random Fields (MRF)
Hidden Markov Models (HMM)
Variational Auto-Encoders (VAE)
Generative Adversarial Networks (GAN)
Normalizing Flows (NF)
Graph Neural Networks (GNN)
Diffusion Probabilistic Models (DPM)
Transformer-based

Figure 1: Classification of the data-driven TDS models included in our study.

- **Class Rebalancing:** Datasets may have a few classes which significantly outnumber others, risking bias towards these dominant classes. For example, a diabetes dataset may contain more non-diabetic than diabetic patient records. This discrepancy leads to prediction models being biased towards the non-diabetic class. TDS can rectify this imbalance by generating additional synthetic diabetic patient records.
- **Dataset Augmentation:** TDS can be used for data augmentation, where the goal is to expand datasets for enhancing ML model robustness and generalization. In our EHR example, this would mean synthesizing records for new patients from all classes.
- **Customized Generation:** The generation of synthetic datasets must sometimes be directed by external factors, in order to create specific scenario data. For instance, in the EHR context, researchers might want to simulate a situation in which certain disease progressions occur more frequently.

Sharing data while preserving privacy may be the only reason for synthesis, but it can also be combined with one of the other purposes. Although tools for class rebalancing or missing value imputation can usually be adapted for data augmentation and vice versa, it is important to choose the right tool for the job. Otherwise, important relationships in the data may not be preserved or the workload and computational costs may increase.

In summary, use cases for TDS basically include all areas in which use-case specific data must be processed or shared and either not enough data are available to achieve high-quality processing results or the available data may not be shared, e.g. for data privacy reasons.

From the perspective of data management, interesting use cases are in particular those in which data management tasks have to be outsourced to an external service provider, but the corresponding instance data has to be kept confidential. Those tasks include schema design [58, 82], query optimization [61, 77], data profiling [1] (e.g., foreign key discovery [16, 113]), and index tuning [106]. Since the execution of these tasks depends heavily on the characteristics of the instance data and the original instance data are not available, the synthesized instance data should compensate for this deficiency.

4 Tabular Data Synthesis Challenges

For all domains of data synthesis, whenever privacy protection is of interest, one challenge is the *Privacy vs. Utility trade-off* [83]. Data utility refers to the ability of the data to serve its intended purpose effectively. Generating synthetic samples while preserving privacy is challenging because enhancing privacy often diminishes the utility of the data, and vice versa [33]. The level of privacy can be achieved using Differential Privacy (DP), a rigorous mathematical framework for guaranteeing privacy in statistical analysis [26].

TDS tools must capture and replicate the main characteristics of the original (real) dataset. This includes the column types and correlations between columns. This is challenging because of the following reasons:

- **Missing values:** Accurately capturing the characteristics of a dataset with information gaps is challenging, as these gaps represent a loss of information and the generated data may poorly reflect the true correlations between the columns of the dataset.
- **Imbalanced datasets:** Capturing the characteristics of minority classes is particularly complicated when these classes are underrepresented. As a consequence of this under-representation, some algorithms may over fit the data, suffer from phenomena such as “mode collapse”, where some classes are not generated at all [37], or generate unrealistic samples of the minority classes [14]. However, for applications

that aim to identify outliers, such as intrusion detection [75], it is very important to generate accurate samples of these minority classes.

- **Diversity of column types:** Unlike images, tabular datasets usually contain a mix of different column types, such as numerical, categorical, temporal, text, or even mixed types consisting of values from different basic types. Different column types might require distinct pre-processing or handling techniques.
- **Complex column distributions:** The distribution of a column contains its spread, tendencies, and patterns in the data, providing valuable insights into its characteristics and relationships with other columns. Capturing complex distributions is challenging because traditional methods such as simply modeling mean and standard deviation may not be sufficient to characterize non-Gaussian distributions.
- **Temporal Dependencies:** The temporal dimension of time series data introduces an additional layer of complexity. Two particular challenges for time series generation are discrete time series, because backpropagation presents problems [9], and long-term dependencies, because their discovery and modeling require extra memory [69].

5 User Needs: TDS Requirements

Due to the wide range of possible applications for tabular data synthesis and the special characteristics of individual datasets, each use case can have very different requirements. Therefore, based on the purposes mentioned in Section 3 and challenges described in Section 4, we have determined a list of twelve possible *functional* and *non-functional* requirements that users may impose on a TDS tool (see Table 1).

The first four functional requirements address (i) the number of non-independent columns the tool can synthesize, (ii) the types and (iii) distributions of columns the tool is able to handle as well as (iv) whether the tool preserves correlations between columns. Early attempts of tabular data synthesis started with basic statistical models, random sampling, and rule-based approaches [8], which can only generate one or two dependent columns at once, or they focused either on only categorical or numerical columns [14]. In addition, many of these attempts used simplified models that assume all columns to be Gaussian distributed. All of the TDS tools examined in this study are capable of working with multi-column datasets. However, they differ in the types of columns, distributions, and correlations they can capture and replicate.

The fifth functional requirement addresses temporal dependencies. State-of-the-art TDS tools that are capable of handling complex column distributions and correlations (e.g., [62,99,118]) do not address temporal dependencies. In contrast, TDS tools developed specifically for time series data differ in whether and how well they can preserve short- and long-term dependencies.

The requirement to preserve integrity constraints refers to the ability of a TDS tool to create a synthetic dataset that does not violate the rules enforced on the original dataset, such as unique column combinations (UCCs) or functional dependencies (FDs) [33]. Inter-table correlations refer to correlations between columns from different tables whose records are linked by foreign keys. While machine learning usually operates on individual tables, complex schemas are the norm in database management. In addition, integrity constraints play a key role in many data management tasks (e.g., FDs in schema design). Inter-table correlations and integrity constraints are therefore particularly important from a data management perspective

The group of non-functional requirements refers to factors that are not directly related to how well the TDS tools capture and replicate the characteristics of a dataset, but to operational properties. This includes factors such as (i) how much configuration is needed before the tools can be properly used, (ii) how much pre-processing of the input data is required so that the tools can process them, (iii) what hardware components (e.g., GPUs) are required for executing the tools, (iv) how much resources (e.g., runtime, memory, electrical power) the tools need, and (v) how well the tools scale with larger datasets.

6 TDS Models and Tool Capabilities

In this section, we give an overview of existing TDS models and tools. We adopt the classification of TDS models into process-driven and data-driven models from [35] and extend it with sub-categories of data-driven models as shown in Figure 1. As explained in Section 1, our focus is on data-driven TDS tools, which synthesize data using a use-case specific (real-world) dataset as input. The following sub-sections provide a description of the tools' underlying models as well as their general strength and weaknesses. The detailed assessment of the tools' performance with respect to the functional requirements is presented in Section 8.

6.1 Imputation-based Models

Many imputation-based TDS tools use either multiple imputation or masking techniques [47, 90, 103]. Multiple imputation is originally a model to handle missing values, where each missing value is replaced by two or more synthetic values [93]. It has two steps: First, it constructs multiple synthetic populations. Then, it draws a random sample from each synthetic population and releases those samples.

Table 1: Potential functional and non-functional requirements for TDS Tools.

	Requirement	**Possible Categories**
Functional	Ability to work with multiple non-independent columns.	Single-column, two-column, or multi-column datasets.
	Ability to handle different types of columns effectively.	Categorical, numerical (continuous and discrete), temporal, text, and mixed (e.g., categorical and numerical).
	Ability to accurately capture and replicate univariate column distributions.	Gaussian and other typical statistical distributions (uniform, exponential, Poisson, binomial, logistic, etc.), skewed, multinomial.
	Ability to preserve correlations between two or more columns.	Joint and conditional probabilities of subsets of columns.
	Ability to preserve temporal dependencies between columns.	Short-term and long-term dependencies.
	Ability to preserve integrity constraints defined at different levels of data granularity (e.g., values, columns, records, tables, or the entire dataset).	Rules or conditions enforced. They can concern values from one record (intra-record) or values from multiple records (inter-record). Examples are unique column combinations (UCCs), functional dependencies (FDs), inclusion dependencies (INDs), or denial constraints (DCs).
	Ability to preserve inter-table correlations.	Parent-child relations and relationships between multiple tables resulting from foreign key references.
Non-functional	Level of configuration the tool needs.	Represents the ability of the tool to synthesize datasets without the need for extensive configuration or fine-tuning. It represents its ”out-of-the-box” capability.
	Level of pre-processing the tool needs.	Represents the need for pre-processing the input data. For example, handling missing values, normalizing columns to similar scales and ranges to support convergence, or encoding columns into a format that can be effectively processed by the tool.
	Hardware the tool needs.	Represents the technical requirements of the TDS, including the need for a GPU for training.
	Resource efficiency of the tool. (time and memory)	Represents the time and memory required to synthesize a dataset.
	Scalability of the tool.	Represents the ability to efficiently handle increasingly large datasets while maintaining high performance and accuracy.

Imputation is easy to understand and implement, and it does not require high computational resources. However, it is highly sensible to bias, it can produce extreme samples, and it can contain several repeats of the observed records [91]. Finally, imputation-based approaches do not model the underlying joint distributions of the real dataset, which means they cannot preserve its semantic integrity [63].

6.2 Sampling Models

Data synthesis is often used to rebalance datasets. The straightforward solution to this problem is the augmentation with additional records of the minority class, known as *random over-sampling*. Accordingly, *random under-sampling* removes records from the majority class. Random over-sampling introduces an increased risk of over-fitting and random under-sampling frequently results in the loss of valuable information intrinsic to the original dataset [8]. Tools such as the Synthetic Minority Over-Sampling Technique (SMOTE) [14], can be used for class rebalancing, but also to generate complete synthetic datasets. The vanilla version only works for continuous data and the synthesized records are linearly dependent on the original minority class records, often leading to over-fitting [8]. Variations of SMOTE address these limitations and still generate samples with low computational resources [62]. The implementation presented in [39] combines the strengths of several other SMOTE variations [24, 40, 96], including the ability to handle categorical columns.

6.3 Discriminative Models

In ML, a distinction is made between *discriminative* and *generative* models. Discriminative models estimate the conditional probability of the output given the input. However, they do not learn the interdependence between all the columns (target and non-target). Discriminative models can be leveraged for TDS using the learned conditional probabilities. One example in privacy-preserving data mining are clustering-based algorithms that generate synthetic data while aiming to maintain certain properties of the original data [70]. However, these models have limitations in handling more intricate data characteristics, because they are built to estimate the probability that an observation belongs to a class and not to learn the complete distribution [31].

6.4 Generative Models

Generative models aim to learn the joint probability distribution of all columns [36]. Therefore, they are suitable for all the purposes introduced in Section 3.

6.4.1 Shallow Generative Models

Shallow generative models have simpler architectures with few or no layers of abstraction or transformation.

We consider **Copula models** and **Probabilistic Graphical models** (PGMs) the most relevant for TDS.

Copulas are mathematical functions that link multivariate joint distributions to their one-dimensional marginal distributions [80]. Simply put, a copula separates the analysis of a multivariate distribution into two parts: the individual behavior of each variable, known as the marginals, and a function that binds the marginals back together, capturing how they relate with each other. Tabular Copula [3] is a Python package that uses Gaussian Copulas [65] to produce synthetic datasets. Its results preserve the statistical properties of the original data. However, as demonstrated in [28], Gaussian copulas can have problems capturing extreme dependency structures in the data.

PGMs represent complex distributions through graphs where nodes represent variables and edges represent probabilistic dependencies between these variables [59]. They are widely used because they explicitly show the dependencies in the data. However, exact inference becomes computationally infeasible with large datasets. Popular PGMs in TDS can be subdivided into Bayesian networks and Markov Random Fields.

Bayesian Networks (BN) are PGMs that represent a set of variables and their conditional dependencies via a directed acyclic graph (DAG). PrivBayes [112] is a popular TDS tool for data privacy. However, it does not preserve data utility and correlations as well as other, more complex tools [74]. BNs are also used to complement other tools, for example to integrate semantic information on the input data into the learning process [114].

Markov Random Fields (MRF) are undirected graphical models, which can model complex interactions and dependencies without assuming a specific direction of influence. One example is PrivMRF [12], which, like PrivBayes, ensures differential privacy, but better preserves data utility. Another example is PrivLava [13], which aims to synthesize tabular data with complex schemas where multiple tables are connected via foreign keys under differential privacy. It also reports better data utility results for single tables than PrivMRF but at the expense of higher computational costs.

6.4.2 Deep Generative Models

Deep Generative Models are composed of multiple neural layers that enable the model to learn hierarchical representations. They leverage deep learning techniques to model the joint probability distribution of a dataset.

Variational Autoencoder (VAEs) [57] learn an encoder network that maps the input data to a latent space and a decoder network that reconstructs the original input from this latent space. Synthetic datasets are generated by sampling new records from the latent space.

In TDS, VAEs preserve the characteristics of the dataset better than sampling and shallow generative models [108]. Nevertheless, VAEs often over-simplify the distributions inherent in the original data because they use a standard Gaussian distribution for the latent space [78]. Additionally, VAEs struggle with discrete or categorical columns because they use a reparametrization trick for backpropagation, which only works well for continuous latent spaces [51].

Examples of TDS tools are tabular VAE (TVAE) [108], discrete VAE [92], which addresses the limitation with discrete columns, DP-VAEGM [15] for differential privacy, and TimeVAE [22] for generating multivariate time series data.

Generative Adversarial Networks (GANs) [37] consist of two main neural networks: a generator and a discriminator. The generator uses random noise as input and generates synthetic data samples, while the discriminator aims to distinguish between real and synthetic samples. During training, the generator and the discriminator are trained in an adversarial manner, with the generator attempting to generate data that fools the discriminator, and the discriminator striving to correctly identify the generator's fake samples. Through this competitive process, and using their implicit modeling of the data distribution, GANs learn to generate realistic and high-quality synthetic data samples that closely resemble the distribution of the real training data [118].

Tools such as medGAN [17], DP-GAN [107], and PATE-GAN [52] were specifically developed for privacy-preserving TDS. However, they sacrifice data utility and report lower performance than a vanilla GAN for many ML tasks [52]. TableGAN [83], TGAN [109], and CT-GAN [108] are able to achieve high data privacy with better data utility. CTGAN also uses a conditional vector to allow controlling the generated classes. Building upon them, the two predominant GAN tools nowadays are CTAB-GAN [117] and its successor CTAB-GAN+ [118]. They can both handle mixed data types, imbalanced datasets, and complex distributions.

GANBLR and GANBLR++ [114, 115] address the fact that GANs are not interpretable and do not exploit any prior knowledge on explicit feature interactions. C3-TGAN [41] introduces mechanisms to preserve explicit attribute correlations and property constraints. Both approaches use Bayesian networks.

Most of the approaches for time series data use Recurrent Neural Networks (RNNs) [94], especially of the type Long Short-Term Memory (LSTM) [46]. TimeGAN [110] combines a GAN model with Autoregressive models (AR) but it chunks the dataset into 24 epochs, which is not adequate for long-term dependencies [69]. DoppelGANger [69] is a custom workflow developed to address the key challenges of time series GAN approaches, such as long-term dependencies, complex multidimensional relationships, mode collapse, and privacy.

Normalizing Flows (NF) use invertible and differentiable transformations to convert simple distributions, such as Gaussians, into complex ones for probabilistic density modeling. This process is flexible and allows exact likelihood estimation but is computationally intensive. For this reason, there are not many NF TDS tools. Durkan et al. [25] demonstrated the effectiveness of NF on tabular and image data synthesis. Yet, Manousakas et al. [73] reports that NF underperform compared to models such as CTGAN [108] and TVAE [108]. Kamthe et al. [54] applied NF to learn the copula density for TDS, effectively capturing relations among columns. However, it performs worse than TVAE [108].

Graph Neural Networks (GNNs) are neural network architectures for processing graph-structured data. They handle irregular data structures using relationships between entities (nodes) and their connections (edges) in a graph. For TDS, records are converted into graph nodes, connected by edges based on their similarity or domain-specific knowledge. This transformation allows GNNs to learn node representations that capture inter-record and inter-column correlations.

GOGGLE [71] is a TDS tool that replaces typical VAE decoder architectures with GNNs. It achieves realistic samples, highlighting the potential of GNNs in the synthesis of complex, domain-aware tabular data. However, GNNs can be computationally and memory-intensive, especially with large graphs [50].

Diffusion Probabilistic Models (DPM) [98] are inspired by non-equilibrium physics and have gained significant importance with the improvements introduced by Yang et al. [100] and Ho et al. [44]. They involve a two-step process where a backward denoising step is trained to remove the noise previously added by a forward diffusion step. DPMs model the data generation process as a reverse diffusion process, where noise is iteratively removed from a random initialization until a sample from the target distribution emerges [44, 98].

DPMs can be classified into three categories: Denoising Diffusion Probabilistic Models (DDPM), Score-based Generative Models (SGM), and Stochastic Differential Equations (SDE). They differ in how they transform noise into data records. DDPMs take a step-by-step approach, gradually refining noise. SGMs use the gradient of the data distribution to directly guide noise towards the outcome. Meanwhile, SDEs treat this transformation as a continuous process, modeling the addition and removal of noise through differential equations.

TDS diffusion tools, such as TabDDPM [62], SOS [56], and STaSy [55], are able to preserve complex distributions and correlations, and are reported to outperform simpler tools, such as TVAE [108] and TableGAN [83] in terms of ML utility. However, they are computationally more expensive than other deep generative alternatives [68]. TSGM [67] is an example for multivariate time series generation using an SGM. It generates records conditioned on past generated observations.

Transformers use an encoder-decoder architecture that revolutionized the field of natural language processing by replacing traditional RNNs and CNNs with attention mechanisms [105]. This allows each element of a sequence to focus on any other elements of the same sequence, effectively capturing long-range dependencies.

Currently, there are a few transformer-based TDS tools, with GReaT [5], REaLTabFormer [99], and TabuLa [116] being notable examples. They all use a pre-trained large language model (LLM) consisting of only a decoder. GReaT uses the LLM GPT-2 and transforms tabular datasets into textual representations before providing them to the LLM (fine-tuning and inference). This step minimizes the required data pre-processing. REaLTabFormer also uses GPT-2 and addresses the generation of synthetic datasets with two tables being in a one-to-many relationship (i.e., one parent and one child table). They aim to reduce extensive fine-tuning, especially for child tables. The authors of TabuLa emphasize that LLM-based TDS tools provide two main advantages: elimination of the need to pre-define column types and elimination of the dimension explosion problem when synthesizing high-dimensional data. However, LLM-based tools have limitations on training efficiency and preserving column correlations [116].

6.5 Probabilistic Database-based Models

Ge et al. [33] remark that most (deep) generation models fail to preserve integrity constraints of the input data in the synthetic output data. To address this issue, they developed a constraint-aware differentially private data synthesis approach called KAMINO. KAMINO preserves denial constraints specified by the user. Similar to VAEs, it first uses the input dataset to learn a latent space and then uses this space to sample the synthetic dataset. The difference is that KAMINO represents this space by a factorized probabilistic database [95] and takes constraint violations into account when sampling the synthetic values one after another. By learning weights for the individual constraints, KAMINO also allows the modeling of soft constraints, which do not strictly have to be fulfilled. Experiments show that KAMINO produces much fewer constraint violations than other privacy-focused approaches, such as PrivBayes [112], DP-VAEGM [15], and PATE-GAN [52]. However, since KAMINO explicitly checks for constraint violations during sampling, it has longer execution times.

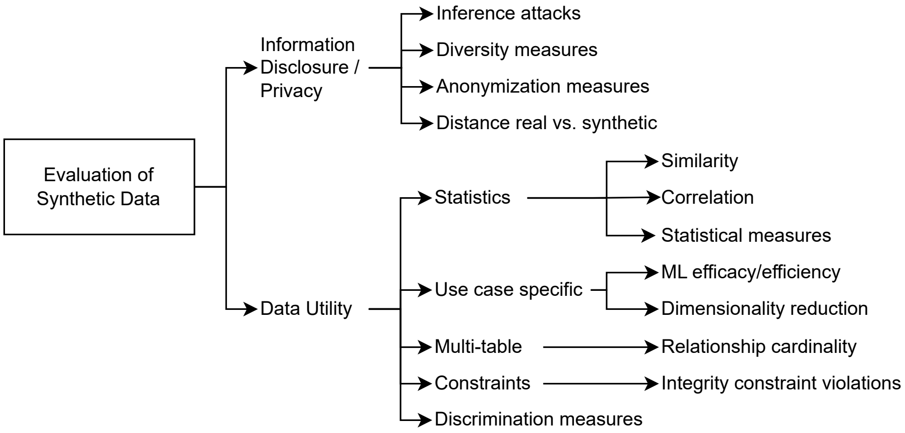

Figure 2: Taxonomy of metrics for evaluating synthetic tabular data identified in our research.

6.6 Hybrid Models & Other Approaches

Some state-of-the-art TDS tools use combinations of different TDS models. Examples of such hybrid tools are AutoDiff [101] and TabSyn [111], which combine VAEs with a diffusion model to improve the performance of typical DDPMs for different column types and distributions. They are also designed to reduce runtime compared to typical diffusion tools.

Gilad et al. [34] proposed a method that uses already-generated tables and connects them via foreign keys while considering cardinality and integrity constraints. In combination with an existing TDS approach, this allows the generation of datasets with complex schemas.

7 Synthetic Data Evaluation Metrics

In contrast to other modalities of synthetic data, such as text or images, the quality of tabular data cannot be easily assessed by human inspection, as its inherent properties (see Section 4), such as column distributions and correlations, are not easily recognizable to humans. In this section, we therefore provide an overview of evaluation metrics typically used in TDS research to measure the quality of synthetic tabular data.

Most of these metrics capture one certain characteristic of the dataset, e.g., whether its correlations are identical to the correlations in the input dataset. However, there is no universally accepted evaluation metric for synthetic data among researchers. This makes comparing the generative capabilities of the different tools difficult, as each work uses its own set of metrics [19].

First attempts, such as TabSynDex [19], aim to provide a universal metric by combining commonly used metrics into a single metric score. However, we argue that a combined metric is only useful if its individual component metrics are appropriate for the purpose for which the synthetic dataset was created (see Section 3).

Goncalves et al. [35] classify TDS evaluation metrics into "data utility" and "information disclosure" metrics, coherent to the *Privacy vs. Data Utility* trade-off discussed in Section 4. Based on this, we classify existing evaluation metrics into further, more detailed, classes. Data utility metrics capture the usefulness of the synthetic dataset and how similar it is compared to its real counterpart. In contrast, information disclosure encompasses all metrics that are related to the privacy aspect of data synthesis. Goncalves et al. describe them as measures of "(...) how much of the real data may be revealed (directly or indirectly) by the synthetic data" [35, p. 6].

In our study, we have compiled a list of metrics used in the publications of the 37 tools we have evaluated. We additionally include metrics employed either in TabSynDex [19], the Synthetic Data Vault [85], or Synthcity [87, 104]. In Figure 2, we present a comprehensive overview of evaluation metrics for synthetic tabular data in the form of a taxonomy. In Table 2, we classify our selected metrics according to this taxonomy and reference examples where they have been used. These examples can serve as a starting point on where to look for a potential implementation. For beginners, it might be beneficial to use libraries such as the sdmetrics [21] from the Synthetic Data Vault [85] or Synthcity [104], as they offer comprehensive functionalities.

In general, users should think carefully about the application of the data to be synthesized and choose their evaluation metrics accordingly. This is especially crucial when handling sensitive data, where protecting privacy takes precedence over data utility. In addition, a metric should always be selected to suit the purpose of the TDS task at hand. For example, when rebalancing the classes of a table, the goal is to change the distribution of some original columns. Thus, using a similarity-based metric for these columns is contradictory.

The most popular metric is ML efficiency, which refers to the application of ML models. For data management, other metrics may be more useful, such as the

Table 2: Classification of evaluation metrics. The references are examples where these metrics are used.

Class	Evaluation Metric
Inference Attack	Categorical Correct Attribute Probability (CAP) [21], Membership Attack [43,83]
Anonymization Measures	K-Anonymization [104], L-Diversity [104], K-Map [104]
Distance Real vs. Synthetic	Novel Row Synthesis [21], Common (leaked) Rows Proportion [104], Distance to closest Record (DCR) [43,83]
Similarity	Range & Outlier Coverage [21], TV Complement [21], Kolmogorow-Smirnow-Test [6,53], Chi-squared Test [6,21], KL Divergence [35], Jensen-Shannon Distance [118], Contingency Similarity [21]
Statistical Measures	Mean, Median, Mode, Variance, Min, Max, %-quantile
Correlation	Pearson Coefficient [21,62,72], Spearman's Coefficient [4,21]
Discrimination Measures	ML Real vs. Synthetic Discrimination [2,104], pMSE-score [19,97]
ML Efficiency	ML Classification [21,35,62,104,108], ML Regression [21,62,104,108]
Dimensionality Reduction	Principal Component Analysis (PCA) [66,76], T-distributed stochastic neighbor embedding (T-SNE) [66,76]
Relationship Cardinality	Cardinality Shape Similarity [21]
IC Violations	g1-Error [33]

statistical similarity of query results, or the correctness of optimized query plans and normalized schemas.

8 Study Results

In our study, we assessed the 37 TDS tools listed in Table 3 on their suitability for the purposes described in Section 3 and their reported performance on the functional requirements listed in Table 1. We have selected these tools based on their popularity, diversity, novelty, and quality (in terms of utility measures such as ML efficiency). The assessment resulted in the matrices shown in Tables 4 and 5. Based on these results, we constructed a tool selection guide helping inexperienced users to select a suitable TDS tool for their specific use case and identified gaps in TDS research.

8.1 Tool Capability Assessment

The first step in our assessment was determining which purposes were addressed by each TDS tool. For example, we consider TDS tools suitable for privacy protection if they include privacy-preserving techniques in their algorithms, as well as privacy evaluation metrics in their experiments. A few tools (e.g., PrivBayes) sacrifice data utility completely in favor of privacy and hence are not really suitable for any other purpose. However, most tools are able to perform missing value imputation, class rebalancing, and data augmentation. The difference between these tools lies in whether they are suitable to protect privacy (either simple[1] or differential privacy [26]), whether they allow customized generation, and which potential user requirements they meet.

Table 3: The 37 TDS tools used in our assessment.

Model	TDS Tool
Sampling	SMOTE [14], Borderline-SMOTE [40], SVM-SMOTE [96], Kmeans-SMOTE [24], Enhanced SMOTE [39], ADASYN [42]
Bayesian Network	PrivBayes [112]
MRF	PrivLava [13], PrivMRF [12]
GAN	medGAN [17], PATE-GAN [52], DTGAN [64], DP-GAN [107], TableGAN [83], TGAN [109], CTGAN [108], C3TGAN [41] CTAB-GAN [118], CTAB-GAN+ [117], GANBLR [114], GANBLR++ [115], TimeGAN [110], DoppelGANger [69]
VAE	TVAE [108], TimeVAE [22], DP-VAEGM [15]
Diffusion (DPM)	TabDDPM [62], TSGM [67], SOS [56], STaSy [55]
Graph NN	GOGGLE [71]
Transformer	GReaT [5], REalTabFormer [99], TabuLa [116]
Prob. Database	KAMINO [33]
Hybrid	AutoDiff [101], TabSyn [111]

Therefore, in the second step we assessed the TDS tools' reported performance on functional requirements. All TDS tools included in our study are able to synthesize multiple dependent columns. Furthermore, we marked the column types which are reported to be effectively handled by the tools, based on the datasets used in their experiments. Some tools only included datasets with either categorical or numerical columns. However, the most complete tools are also able to work with temporal, text, and mixed column types. In addition, we marked whether the tools are reported to perform well with complex distributions or whether they assume columns to be Gaussian distributed. As some of the oldest tools focus on privacy protection and do not aim to preserve column correlations, we also marked whether or not the tools preserve such correlations. Similarly, for tools specially designed to handle time series datasets, we marked whether they address short-term and/or long-term dependencies.

Finally, we assessed whether the TDS tools address challenges outside the ML community, such as integrity constraints and inter-table correlations. For inter-table correlations, we checked whether the tools address two (e.g., a parent-child relationship) or more tables.

During our assessment, we observed a lack of information published with respect to the non-functional

[1] We consider a privacy-preserving technique to be simple if it cannot provide a concrete privacy guarantee, as in the case of differential privacy, but it ensures that the generated records do not exactly resemble the original records, which is usually evaluated by measuring their distance (see DCR in Table 2).

Table 4: Assessment of the 37 TDS tools included in this study on their suitability for the different purposes.

	Privacy		Missing Value	Class	Dataset	Customized
Tools	**Simple**	**Differential**	**Imputation**	**Rebalancing**	**Augmentation**	**Generation**
Enhanced SMOTE*			×	×	×	
ADASYN				×		
PrivBayes		×				
PrivMRF		×	×	×	×	
PrivLava		×	×	×	×	×
medGAN	×					
PATE-GAN		×				
DP-GAN		×				
TableGAN	×		×	×	×	
TGAN	×		×	×	×	
DTGAN		×	×	×	×	
CTGAN	×		×	×	×	×
C3TGAN			×	×	×	×
CTAB-GAN	×		×	×	×	×
CTAB-GAN+		×	×	×	×	×
GANBLR			×	×	×	×
GANBLR++			×	×	×	×
TimeGAN			×	×	×	
DoppelGANger	×		×	×	×	
TVAE			×	×	×	
TimeVAE			×	×	×	
DP-VAEGM		×	×	×	×	
TabDDPM	×		×	×	×	×
TSGM			×	×	×	
SOS			×	×	×	
STaSy			×	×	×	
GOGGLE			×	×	×	
GReaT			×	×	×	×
REaLTabFormer	×		×	×	×	×
TabuLa			×	×	×	×
KAMINO		×	×	×	×	
AutoDiff	×		×	×	×	×
TabSyn	×		×	×	×	×

*Enhanced SMOTE serves as a representative of all SMOTE variations

Table 5: Assessment of the 37 TDS tools on their reported performance on the functional requirements.

	Column Types						Complex	Correlations		Temporal		Integrity Constraints	
Tools	**Categorical**	**Num. Continuous**	**Num. Discrete**	**Temporal**	**Text**	**Mixed Cat./Num.**	**Distributions**	**Intra-table**	**Inter-table**	**Short**	**Long**	**Intra-record**	**Inter-record**
Enhanced SMOTE*	×	×	×										
ADASYN		×											
PrivBayes	×							×					
PrivMRF	×	×	×					×					
PrivLava	×	×	×				×	×	×				
medGAN	×	×	×					×					
PATE-GAN	×	×	×										
DP-GAN	×	×	×										
TableGAN	×	×	×					×					
TGAN	×	×	×					×					
DTGAN	×	×	×					×					
CTGAN	×	×	×				×	×					
C3TGAN	×	×	×					×				×	
CTAB-GAN	×	×	×			×	×	×					
CTAB-GAN+	×	×	×			×	×	×					
GANBLR	×						×	×					
GANBLR+	×	×	×				×	×					
TimeGAN	×	×	×	×				×		×			
DoppelGANger	×	×	×	×			×	×		×	×		
TVAE	×	×	×				×	×					
TimeVAE	×	×	×	×				×		×			
DP-VAE-GM	×	×	×					×					
TabDDPM	×	×	×				×	×					
TSGM	×	×	×	×				×		×			
SOS	×	×	×				×	×					
STaSy	×	×	×				×	×					
GOGGLE	×	×	×				×	×					
GReaT	×	×	×		×	×	×	×					
REaLTabFormer	×	×	×	×	×	×	×	×	×				
TabuLa	×	×	×		×	×	×	×					
KAMINO	×	×	×			×		×				×	×
AutoDiff	×	×	×			×	×	×					
TabSyn	×	×	×			×	×	×					

*Enhanced SMOTE serves as a representative of all SMOTE variations

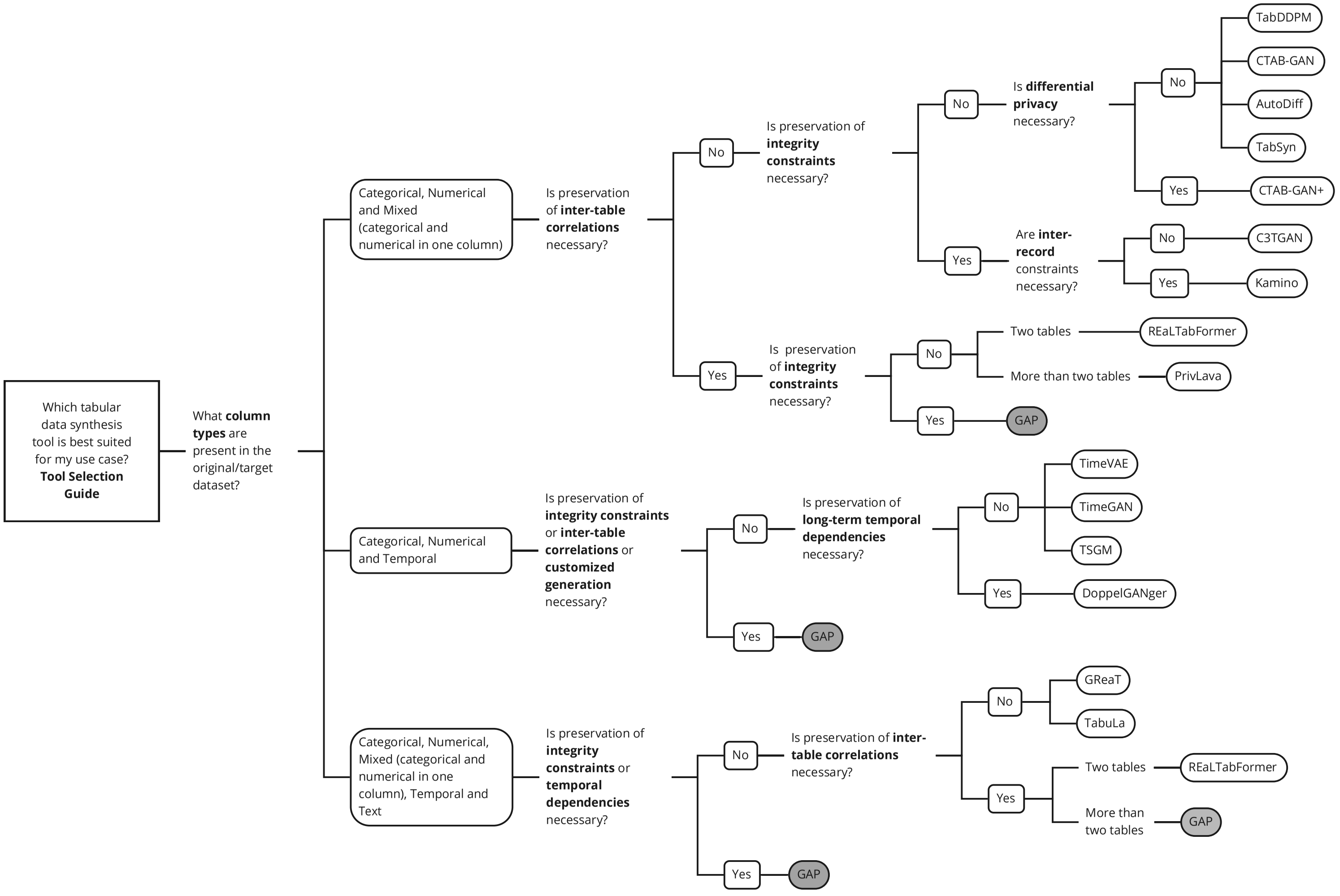
Which tabular data synthesis tool is best suited for my use case? Tool Selection Guide
What column types are present in the original/target dataset?
Categorical, Numerical and Mixed (categorical and numerical in one column)
Is preservation of inter-table correlations necessary?
No
Is preservation of integrity constraints necessary?
No
Is differential privacy necessary?
No
TabDDPM
CTAB-GAN
AutoDiff
TabSyn
Yes
CTAB-GAN+
Yes
Are inter-record constraints necessary?
No
C3TGAN
Yes
Kamino
Yes
Is preservation of integrity constraints necessary?
No
Two tables
REaLTabFormer
More than two tables
PrivLava
Yes
GAP
Categorical, Numerical and Temporal
Is preservation of integrity constraints or inter-table correlations or customized generation necessary?
No
Is preservation of long-term temporal dependencies necessary?
No
TimeVAE
TimeGAN
TSGM
Yes
DoppelGANger
Yes
GAP
Categorical, Numerical, Mixed (categorical and numerical in one column), Temporal and Text
Is preservation of integrity constraints or temporal dependencies necessary?
No
Is preservation of inter-table correlations necessary?
No
GReaT
TabuLa
Yes
Two tables
REaLTabFormer
More than two tables
GAP
Yes
GAP

Figure 3: Tool selection guide resulting from the assessment of 37 TDS tools, based on their reported performance on the functional requirements identified in Section 5.

requirements, such as the level of customization, preprocessing, and hardware required, as well as the resource efficiency and scalability of the tools.

8.2 TDS Tool Selection Guide

Based on the assessments results, we consolidated all factors into a tool selection guide, which is shown in Figure 3 in the form of a decision tree. As the complete guide with all 37 tools and all selection factors was too large for a printed presentation, we removed the simplest branches from this tree. These include: (i) TDS tools that can only be used to address the purposes of missing value imputation, class rebalancing, or data augmentation, (ii) TDS tools that can handle only categorical or numerical columns, (iii) TDS tools that do not effectively preserve column correlations, and (iv) TDS tools that assume all column distributions to be Gaussian.

The selection guide comprises five node levels and ends with leaf nodes that either are a suitable TDS tool for the use case or a "Gap-Leaf" that represents a research gap. The questions of the individual branches are meant to show users the differences between the TDS tools. Given the limited amount of information available on the tools' resource efficiency, the guide suggests tools that cover the functional requirements of the respective branch and avoids recommending overly complex tools.

The selection guide allows users to assess the suitability of a TDS tool for their use case by answering questions about their own dataset and intended purpose. Since it requires no expertise about the tools' underlying models (e.g., GAN or MRF), this approach is much easier to use than navigating the selection process based on the differences between those models. If the selection process ends up with a "Gap-Leaf", the users can still identify the nearest possible tool and its limitations. For example, if their dataset includes categorical, numerical, temporal, and text columns, and the preservation of integrity constraints is necessary, the users will find that there are no suitable TDS tools to date. However, they can see that (i) REaLTabFormer [99] works for all those column types, but does not preserve integrity constraints and (ii) KAMINO [33] preserves integrity constraints, but does not support temporal and text columns.

8.3 TDS Research Gaps

Our tool selection guide reveals two primary research gaps, both of which are particularly relevant to the data management community. The first is the synthesis of datasets with complex relational schemas consisting of multiple tables linked by foreign keys. The second is the preservation of integrity constraints.

PrivLava [13] is a first promising approach to synthesizing datasets with complex schemas. Its biggest disadvantage is that it is restricted to non-cyclic reference graphs. Since such cycles are not uncommon in large schemas, especially self-references (e.g., one person refers to another), this is a significant limitation.

KAMINO [33] and C3TGAN [41] provide initial solutions for preserving integrity constraints when synthesizing single tables. While KAMINO is able to preserve denial constraints (including UCCs and FDs), C3TGAN achieves the same for simple constraints that are limited to the values of individual records (e.g., age > experience). Constraints that are not covered by either approach are, for example, arithmetic constraints (e.g., gross = net + tax) and cardinality restrictions (e.g., a person is only allowed to make a maximum of five debits per day). Finally, there is currently no tool that supports both complex schemas and integrity constraints.

In addition to these gaps, there is a need for evaluation metrics related to data management tasks (analogous to ML efficiency) and benchmarking frameworks that help users to identify the tools most useful for their use case.

9 Conclusion

Data scarcity and data privacy have become fundamental problems for data-driven models across application domains [11]. While data synthesis tools are already used to mitigate these issues, choosing the right tool for a given use case has become increasingly complex.

In this paper, we provided an overview of the challenges and solutions that currently exist in the field of tabular data synthesis (TDS). To enable a systematic tool selection, we first identified functional and non-functional requirements of potential use cases and then evaluated 37 TDS tools with regard to these requirements. Based on the two resulting assessment matrices, we developed a decision guide that supports users in selecting the right tool for their specific use case and identified open challenges in current TDS research.

One key finding is that TDS research has so far focused on the ML community and that important data management requirements have not received enough attention. This applies in particular to the generation of datasets with complex schemas and the preservation of integrity constraints.

In our future work, we plan to develop a benchmarking framework that enables automatic evaluation of a tool's fitness for use for different real-world use cases. In this way, users can be guided by experimental results rather than reported performance numbers.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 495170629 and the Federal Ministry for Economic Affairs and Climate Action (BMWK) - 01MD22001D.

10 References

[1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. Profiling relational data: a survey. *VLDB Journal*, 24(4):557–581, 2015.

[2] Moustafa Alzantot, Supriyo Chakraborty, and Mani B. Srivastava. Sensegen: A deep learning architecture for synthetic sensor data generation. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 188–193, 2017.

[3] BiomedDAR. Tabular copula. https://biomeddar.github.io/copula-tabular/, Jan 2023. Accessed: 2024-10-30.

[4] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024.

[5] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–18, 2023.

[6] Stavroula Bourou, Andreas El Saer, Terpsichori Helen Velivassaki, Artemis C. Voulkidis, and Theodore B. Zahariadis. A review of tabular data synthesis using GANs on an IDS dataset. *Information*, 12(9):375, 2021.

[7] Kendrick Boyd. Create Synthetic Time-series Data with DoppelGANger and PyTorch. https://gretel.ai/blog/create-synthetic-time-series-with-doppelganger-and-pytorch, 2022. Gretel AI. Accessed: 2024-10-30.

[8] Jakob Brandt and Emil Lanzen. A comparative review of SMOTE and ADASYN in imbalanced data classification. Master's thesis, Uppsala Universitet, 2020.

[9] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. Generative adversarial networks in time series: A systematic literature review. *ACM Computing Surveys*, 55(10):199:1–199:31, 2023.

[10] Nicolas Bruno and Surajit Chaudhuri. Flexible database generators. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 1097–1107, 2005.

[11] Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Sina Noack, Hendrik Patzlaff, Hazar Harmouch, and Felix Naumann. The effects of data quality on ml-model performance. *CoRR*, abs/2207.14529, 2022.

[12] Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. Data synthesis via differentially private markov random fields. *Proceedings of the VLDB Endowment (PVLDB)*, 14(11):2190–2202, 2021.

[13] Kuntai Cai, Xiaokui Xiao, and Graham Cormode. PrivLava: Synthesizing relational data with foreign keys under differential privacy. *Proceedings of the ACM on Management of Data*, 1(2):142:1–142:25, 2023.

[14] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[15] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaafar, and Haojin Zhu. Differentially private data generative models. *CoRR*, abs/1812.02274, 2018.

[16] Zhimin Chen, Vivek R. Narasayya, and Surajit Chaudhuri. Fast foreign-key detection in microsoft SQL server powerpivot for excel. *Proceedings of the VLDB Endowment (PVLDB)*, 7(13):1417–1428, 2014.

[17] Edward Choi, Siddharth Biswal, Bradley A. Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the Machine Learning for Health Care Conference (MLHC)*, pages 286–305, 2017.

[18] Peter Christen and Agus Pudjijono. Accurate synthetic generation of realistic personal information. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 507–514, 2009.

[19] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang. A universal metric for robust evaluation of synthetic tabular data. *IEEE Transactions on Artificial Intelligence*, 5(1):300–309, 2024.

[20] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[21] Inc. DataCebo. Synthetic Data Metrics. https://docs.sdv.dev/sdmetrics/, Dec 2023. Version 0.13.0. Accessed: 2024-10-30.

[22] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *CoRR*, abs/2111.08095:1–10, 2021.

[23] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8780–8794, 2021.

[24] Georgios Douzas, Fernando Bação, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465:1–20, 2018.

[25] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7509–7520, 2019.

[26] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12, 2006.

[27] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. Relational data synthesis using generative adversarial networks: A design space exploration. *Proceedings of the VLDB Endowment (PVLDB)*, 13(11):1962–1975, 2020.

[28] Jean-David Fermanian and Olivier Scaillet. Some statistical pitfalls in copula modeling for financial applications. *Capital Formation, Governance and Banking*, pages 1–24, 2004.

[29] Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, 10(15):1–41, 2022.

[30] João Fonseca and Fernando Bação. Tabular and latent space synthetic data generation: A literature review. *Journal of Big Data*, 10(1):115, 2023.

[31] David Foster. *Generative Deep Learning*. O'Reilly Media, Inc., Sebastopol, California, USA, 1st edition, 2019.

[32] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database systems - the complete book (2. ed.)*. Pearson Education, 2009.

[33] Chang Ge, Shubhankar Mohapatra, Xi He, and Ihab F. Ilyas. Kamino: Constraint-aware differentially private data synthesis. *Proceedings of the VLDB Endowment (PVLDB)*, 14(10):1886–1899, 2021.

[34] Amir Gilad, Shweta Patwa, and Ashwin Machanavajjhala. Synthesizing linked data under cardinality and integrity constraints. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 619–631, 2021.

[35] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC Medical Research Methodology*, 20:108, 2020.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 2016.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[38] Jim Gray, Prakash Sundaresan, Susanne Englert, Kenneth Baclawski, and Peter J. Weinberger. Quickly generating billion-record synthetic databases. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 243–252, 1994.

[39] Christos K. Aridas Guillaume Lemaître, Fernando Nogueira. Smote in imbalanced-learn. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE, 2023. Accessed: 2024-10-30.

[40] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the International Conference on Intelligent Computing (ICIC)*, pages 878–887, 2005.

[41] Peiyi Han, Wen Xu, Wanyu Lin, Jiahao Cao, Chuanyi Liu, Shaoming Duan, and Haifeng Zhu. C3-TGAN - controllable tabular data synthesis with explicit correlations and property constraints. *TechRxiv*, pages 1–12, 2023.

[42] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328, 2008.

[43] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[44] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1–12, 2020.

[45] Joseph E. Hoag and Craig W. Thompson. A parallel general-purpose synthetic data generator. *SIGMOD Record*, 36(1):19–24, 2007.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[47] James Honaker, Gary King, and Matthew Blackwell. Amelia ii: A program for missing data. *Journal of Statistical Software*, 45(7):1–47, 2011.

[48] Kenneth Houkjær, Kristian Torp, and Rico Wind. Simple and realistic data generation. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 1243–1246, 2006.

[49] MOSTLY AI Inc. Data Access and Data Insights for Everyone. https://mostly.ai/, 2023. Accessed: 2024-10-30.

[50] Satoshi Iwata, Remzi H. Arpaci-Dusseau, and Akihiko Kasagi. An analysis of graph neural network memory access patterns. In *Proceedings of the Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 914–921, 2023.

[51] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2017.

[52] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: generating synthetic data with differential privacy guarantees. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–21, 2019.

[53] Frank J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[54] Sanket Kamthe, Samuel Assefa, and Marc Peter Deisenroth. Copula flows for synthetic data generation. *CoRR*, abs/2101.00598:1–15, 2021.

[55] Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–27, 2023.

[56] Jayoung Kim, Chaejeong Lee, Yehjin Shin, Sewon Park, Minjung Kim, Noseong Park, and Jihoon Cho. SOS: score-based oversampling for tabular data. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 762–772, 2022.

[57] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

[58] Henning Köhler and Sebastian Link. SQL schema design: foundations, normal forms, and normalization. *Information Systems*, 76:88–113, 2018.

[59] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, Cambridge, MA, USA, 2009.

[60] Heejoon Koo and To Eun Kim. A comprehensive survey on generative diffusion models for structured data. *CoRR*, abs/2306.04139:1–20, 2023.

[61] Jan Kossmann, Thorsten Papenbrock, and Felix Naumann. Data dependencies for query optimization: A survey. *VLDB Journal*, 31(1):1–22, 2022.

[62] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 17564–17579, 2023.

[63] Peter Kowalczyk, Giacomo Welsch, and Frédéric Thiesse. Towards a taxonomy for the use of synthetic data in advanced analytics. *CoRR*, abs/2212.02622, 2022.

[64] Aditya Kunar, Robert Birke, Zilong Zhao, and Lydia Y. Chen. DTGAN: differential private training for tabular GANs. *CoRR*, abs/2107.02521:1–17, 2021.

[65] David Xianglin Li. On default correlation: A copula function approach. *The Journal of Fixed Income*, 9(4):43–54, 2000.

[66] Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. TTS-GAN: A transformer-based time-series generative adversarial network. In *Proceedings of the International Conference on Artificial Intelligence in Medicine (AIME)*, pages 133–143, 2022.

[67] Haksoo Lim, Minjung Kim, Sewon Park, and Noseong Park. Regular time-series generation using SGM. *CoRR*, abs/2301.08518:1–12, 2023.

[68] Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time-series applications: A survey. *Frontiers of Information Technology and Electronic Engineering*, 25(1):19–41, 2024.

[69] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. Using GANs for sharing networked time series data: Challenges, initial promise, and open questions. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 464–483, 2020.

[70] Kun Liu, Chris Giannella, and Hillol Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 297–308, 2006.

[71] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. GOGGLE: generative modelling for tabular data by learning relational structure. In *Proceedings of the*

International Conference on Learning Representations (ICLR), pages 1–22, 2023.

[72] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical evaluation on synthetic data generation with generative adversarial network. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS)*, pages 16:1–16:6, 2019.

[73] Dionysis Manousakas and Sergül Aydöre. On the usefulness of synthetic tabular data generation. *CoRR*, abs/2306.15636:1–12, 2023.

[74] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4435–4444, 2019.

[75] Marija S. Milosevic and Vladimir M. Ciric. Extreme minority class detection in imbalanced data for network intrusion. *Computers & Security*, 123:102940, 2022.

[76] Khanh-Hoi Le Minh and Kim-Hung Le. Airgen: GAN-based synthetic data generator for air monitoring in smart city. In *International Forum on Research and Technology for Society and Industry (RTSI)*, pages 317–322, 2021.

[77] B. Muthuswamy and Larry Kerschberg. A detailed database statistics model for realtional query optimization. In *Proceedings of the ACM Annual Conference on the Range of Computing*, pages 439–448, 1985.

[78] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4723–4732, 2019.

[79] Felix Naumann, Johann Christoph Freytag, and Ulf Leser. Completeness of integrated information sources. *Information Systems*, 29(7):583–615, 2004.

[80] Roger B Nelsen. *An introduction to Copulas*. Springer Series in Statistics, Portland, OR, USA, 2nd edition, 2006.

[81] Andrea Neufeld, Guido Moerkotte, and Peter C. Lockemann. Generating consistent test data for a variable set of general consistency constraints. *VLDB Journal*, 2(2):173–213, 1993.

[82] Thorsten Papenbrock and Felix Naumann. Data-driven schema normalization. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 342–353, 2017.

[83] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment (PVLDB)*, 11(10):1071–1083, 2018.

[84] European Parliament. Boosting data sharing in the EU: what are the benefits? https://www.europarl.europa.eu/news/en/headlines/ society/20220331STO26411/boosting-data-sharing-in-the-eu-what-are-the-benefits, Nov 2023. Accessed: 2024-10-30.

[85] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.

[86] Vamsi K. Potluru, Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreacic, Ganapathy Mani, Saheed Obitayo, Deepak Paramanand, Natraj Raman, Mikhail Solonin, Srijan Sood, Svitlana Vyetrenko, Haibei Zhu, Manuela Veloso, and Tucker Balch. Synthetic data applications in finance. *CoRR*, abs/2401.00081, 2024.

[87] Zhaozhi Qian, Robert Davis, and Mihaela van der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[88] Tilmann Rabl and Meikel Poess. Parallel data generation for performance analysis of large, complex RDBMS. In *Proceedings of the International Workshop on Testing Database Systems (DBTest)*, page 5, 2011.

[89] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

[90] Trivellore E. Raghunathan, Jerome P. Reiter, and Donald B. Rubin. Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics*, 19:1–16, 2003.

[91] Jerome P. Reiter. Satisfying disclosure restrictions with synthetic data sets. *Journal of Official Statistics*, 18:1–13, 2002.

[92] Jason Tyler Rolfe. Discrete variational autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–33, 2017.

[93] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley and Sons, Inc., New York, 1987.

[94] David E. Rumelhart, Geoffrey Hinton, and

Ronald Williams. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.

[95] Christopher De Sa, Ihab F. Ilyas, Benny Kimelfeld, Christopher Ré, and Theodoros Rekatsinas. A formal framework for probabilistic unclean databases. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 6:1–6:18, 2019.

[96] Hartayuni Sain and Santi Wulan Purnami. Combine sampling support vector machine for imbalanced data classification. *Procedia Computer Science (The Third Information Systems International Conference)*, 72:59–66, 2015.

[97] Joshua Snoke, Gillian M. Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(3):663–688, 2018.

[98] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the JMLR Workshop of the International Conference on Machine Learning (ICML)*, pages 2256–2265, 2015.

[99] Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *CoRR*, abs/2302.02041:1–17, 2023.

[100] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 11895–11907, 2019.

[101] Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Merhdad Honarkhah, and Guang Cheng. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. *CoRR*, abs/2310.15479:1–12, 2023.

[102] European Union. Regulation (EU) 2022/868 of the European Parliament and of the Council. https://eur-lex.europa.eu/eli/reg/2022/868/oj, May 2022. Accessed: 2024-10-30.

[103] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011.

[104] The van der Schaar Lab. Synthcity: A library for generating and evaluating synthetic tabular data. https://github.com/vanderschaarlab/synthcity, 2023. Accessed: 2024-10-30.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[106] Yang Wu, Xuanhe Zhou, Yong Zhang, and Guoliang Li. Automatic index tuning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2024.

[107] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739:1–9, 2018.

[108] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7333–7343, 2019.

[109] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. *CoRR*, abs/1811.11264:1–12, 2018.

[110] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5509–5519, 2019.

[111] Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[112] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems*, 42(4):25:1–25:41, 2017.

[113] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc, and Divesh Srivastava. On multi-column foreign key discovery. *Proceedings of the VLDB Endowment (PVLDB)*, 3(1):805–814, 2010.

[114] Yishuo Zhang, Nayyar Abbas Zaidi, Jiahui Zhou, and Gang Li. GANBLR: A tabular data generation model. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 181–190, 2021.

[115] Yishuo Zhang, Nayyar Abbas Zaidi, Jiahui Zhou, and Gang Li. GANBLR++: incorporating

capacity to generate numeric attributes and leveraging unrestricted bayesian networks. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 298–306, 2022.

[116] Zilong Zhao, Robert Birke, and Lydia Y. Chen. Tabula: Harnessing language models for tabular data synthesis. *CoRR*, abs/2310.12746:1–9, 2023.

[117] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. CTAB-GAN: effective table data synthesizing. In *Proceedings of the Asian Conference on Machine Learning (ACML)*, pages 97–112, 2021.

[118] Zilong Zhao, Aditya Kunar, Robert Birke, Hiek Van der Scheer, and Lydia Y. Chen. CTAB-GAN+: enhancing tabular data synthesis. *Frontiers in Big Data*, 6, 2024.