Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Special Track on AI for Good

A Novel GAN Approach to Augment Limited Tabular Data for Short-Term Substance Use Prediction

Nguyen Thach[1] , **Patrick Habecker**[1] , **Bergen Johnston**[1] , **Lillianna Cervantes**[1] , **Anika Eisenbraun**[1] , **Alex Mason**[1] , **Kimberly Tyler**[1] , **Bilal Khan**[2] and **Hau Chan**[1]
[1]University of Nebraska-Lincoln
[2]Lehigh University
nate.thach@huskers.unl.edu, {phabecker2, bergenjohnston, lcervantes4, aeisenbraun3, amason19, ktyler2, hchan3}@unl.edu, bik221@lehigh.edu

Abstract

Substance use is a global issue that negatively impacts millions of persons who use drugs (PWUDs). In practice, identifying vulnerable PWUDs for efficient allocation of appropriate resources is challenging due to their complex use patterns (e.g., their tendency to change usage within months) and the high acquisition costs for collecting PWUD-focused substance use data. Thus, there has been a paucity of machine learning models for accurately predicting short-term substance use behaviors of PWUDs. In this paper, using longitudinal survey data of 258 PWUDs in the U.S. Great Plains collected by our team, we design a novel GAN that deals with high-dimensional low-sample-size tabular data and survey skip logic to augment existing data to improve classification models' prediction on (A) whether the PWUDs would increase usage and (B) at which ordinal frequency they would use a particular drug within the next 12 months. Our evaluation results show that, when trained on augmented data from our proposed GAN, the classification models improve their predictive performance (AUROC) by up to 13.4% in Problem (A) and 15.8% in Problem (B) for usage of marijuana, meth, amphetamines, and cocaine, which outperform state-of-the-art generative models.

1 Introduction

Substance use can create short- and long-term negative consequences for persons who use drugs (PWUDs) [HHS, 2023; Newcomb and Bentler, 1989; NIDA, 2020]. These consequences include mental illness, HIV/AIDS, hepatitis, drug overdose, and death. Within the U.S. alone, an estimated 161.8 million people aged 12 or older used a substance (out of which 40.0 million used an illicit drug) in the past month before being interviewed in 2021 [SAMHSA, 2022]. Furthermore, according to [NIDA, 2023], substance-involved overdose deaths, including those related to illicit drugs and prescription opioids, continue to increase over the years, with 106,699 deaths in 2021 compared to 91,799 (+16%) in 2020 and 70,630 (+51%) in 2019. This alarming trend also applies globally, with the estimated number of PWUDs in the past 12 months reaching 296 million in 2021 from 240 million in 2011 (out of which 39.5 million and 27.3 million had drug use disorders, respectively) [United Nations, 2023b].

In response to these large-scale negative impacts, various public and private organizations around the globe have prompted initiatives for preventing and reducing substance use at both population and individual levels. Prominent examples are the United Nation's Sustainability Goal 3: "Strengthen the prevention and treatment of substance abuse" [United Nations, 2023a] and U.S. Department of Health and Human Services (HHS)'s Healthy People 2030: "Reduce misuse of drugs and alcohol" [ODPHP, 2020].

Generally, approaches toward these initiatives focus on designing and deploying intervention and outreach programs/resources (e.g., rehabs and consulting services) for PWUDs, with the main goal of reducing and eliminating their usage of certain substances [Ray *et al.*, 2020; Colledge-Frisby *et al.*, 2023]. While these programs and resources have shown to be effective to some extent [Ouimette *et al.*, 1997; Tanner-Smith *et al.*, 2016], they often require volunteer participation from PWUDs, who face the difficulties and reluctances of (self-)evaluating and (self-)determining whether they want or need help [Russell *et al.*, 2021; Wilson and Brown, 2023]. Even when PWUDs agree to use these programs/resources, they may have already experienced prior harms such as overdose and mental illness [SAMHSA, 2021; Andersson *et al.*, 2023]. Therefore, it is important to prevent harm from occurring in the first place by carefully identifying PWUDs at the highest risk (i.e., those who are prone to drastically increase usage of some drug) and allocating them appropriate resources to reduce or eliminate potential harms.

Unfortunately, forecasting individual substance use behavior is challenging due to its complex patterns (e.g., drug use frequency and co-use of multiple drugs) and tendency to change over time at short timescales (i.e., within months) [Karamouzian *et al.*, 2022; Linden-Carmichael *et al.*, 2022; Lorvick *et al.*, 2023], as well as the lack of appropriate data and models for predicting short-term future substance use (see **Existing Efforts and Limitations** and additional contents in the full paper attached in the supplementary material).

Thus, our goal is to design accurate predictive models for modeling short-term drug usage (i.e., within months) from PWUDs to aid healthcare agencies, local communities, policymakers, and other stakeholders in the efficient allocation of

resources to PWUDs who need the most help. An effective predictive model has the potential to improve the well-being of millions of PWUDs and hampers the ongoing rapid growth of drug use prevalence and drug overdose death rates.

Our Approach and Associated Challenges. To address the above shortcomings of substance use data and models, we formed a collaboration between computer scientists and domain experts in substance use research, including social scientists, intervention specialists, and survey interviewers from University of Nebraska-Lincoln and Rural Drug Addiction Research Center. With IRB approval, our team recruited a local sample of 258 PWUDs in the Great Plains of the U.S. from which longitudinal survey data were collected and stored in tabular format (detailed below in Section 2.1). Despite its high value and relevance to our stated goal, the sample size is small due to various challenges during data collection (mainly disruptions due to COVID-19 and the lack of visible harm reduction movements and punitive laws in the region), As a result, our preliminary predictive models trained on currently available data achieved subpar performance with respect to the baseline (demonstrated in our full paper). These models seem to overfit due to the limited training examples ($<$200) with respect to the number of (preprocessed) features ($>$600).

One effective way to tackle overfitting on small datasets is via data augmentation [Shorten and Khoshgoftaar, 2019] i.e., creating synthetic samples based on real data to increase its sample size. *Deep generative models* such as the popular Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014] provide powerful tools for this purpose due to their flexibility in representing complex and high-dimensional data distributions. In recent years, these models have been extended to generate tabular data, especially healthcare records, using specialized architectures (e.g., GOGGLE [Liu *et al.*, 2022]) and/or novel training algorithms (CTGAN [Xu *et al.*, 2019]), which in turn achieved promising performance across different evaluation criteria.

However, these models were benchmarked mostly on moderate to large datasets that typically contain at least thousands of training examples. In rare cases when being applied on small datasets with less than 1000 samples, e.g., Diabetes (768) and Breast (569) [Street *et al.*, 1993] as reported respectively in the work for GOGGLE and TabDDPM [Kotelnikov *et al.*, 2023], the proposed models and its selected comparatives often either fail to or narrowly outperform simple baselines such as Bayesian networks [Pearl, 2011] and SMOTE [Chawla *et al.*, 2002] in terms of their considered evaluation metrics. The benchmark datasets for state-of-the-art models also contain no more than 200 features, which is substantially smaller than the size of our feature set ($>$600 after preprocessing). Additionally, the survey used for collecting our tabular data contains *skip logic*, a commonly employed functionality in survey design for social science applications [Fowler, 1995; Dillman *et al.*, 2014], which has not been addressed by the aforementioned state-of-the-art models.

Our Contributions. In this paper, we design a novel specialized generative model to augment our tabularized survey data with skip logic in order to improve the predictive performance of our classification models, which ultimately predict the following two short-term substance use behaviors: for a given PWUD and a certain drug, (A) whether they would increase its usage and (B) at which frequency (on a pre-defined ordinal scale) they would use it within the next 12 months. We summarize our contributions as follows:

(I) We believe ours is the first work that addresses *skip logic* from surveys in tabular data generation. We demonstrate its practical value by showing both conceptually and empirically how enforcing constraints from skip logic (or *skip constraints* for brevity) positively affects the training of our generative model. We are also one of the first to investigate the real-world feasibility of deep generative models in settings where the number of features exceed the sample size i.e., high-dimension low-sample-size (HDLSS). (See **Related Work** in our full paper.)

(II) We design a novel GAN that deals with small tabular data containing $<$258 samples and 210 features (209 plus one target variable). Specifically, we leverage CTGAN [Xu *et al.*, 2019], a well-known tabular GAN, by incorporating an auxiliary classifier [Park *et al.*, 2018; Zhao *et al.*, 2022] within its architecture to generate high-quality samples conditioned on the corresponding target variable. Since the transformed feature space has high dimensionality (over 600), we also embed a global feature selection mechanism while training the auxiliary classifier's by employing the novel approach from [Margeloiu *et al.*, 2023] that was shown to perform well on even smaller and higher dimensional data than ours. Finally, to enforce the skip constraints stated in (i), we take advantage of CTGAN's built-in conditional vector.

(III) We implement and train the proposed GAN and use it to augment our (training) data. The augmented data is then used to train binary and multiclass classification models for predictive problems (A) and (B), respectively, for each of the following drugs: marijuana, methamphetamine, amphetamines, and cocaine. Our experimental results show that the average Area under the Receiver Operating Characteristic curve (AUROC) evaluated on multiple distinct sets of test data is improved by up to 13.4% in Problem (A) and 15.8% in Problem (B) when the data is augmented, which is significantly higher than what yielded using state-of-the-art generative models.

2 Problem Description

2.1 Background

Our Tabular Data. The recruitment of PWUDs started in 2019 under the respondent-driven sampling scheme [Heckathorn, 2014] in the Great Plains of the U.S. and has continued to the present time. Enrolled PWUDs were followed up within 4–12 months after their initial visit and took the same survey as before. In the survey, each PWUD answered questions on a computer regarding their individual attributes, including the drug use behavior of 18 different drugs. Use frequencies of considered drugs (e.g., marijuana, cocaine, amphetamines, and methamphetamine) were inquired on an ordinal scale (1–8) of {never, less than once a month, once a month, once a week, 2–6 times a week, once a day,

2–3 times a day, 4 or more times a day}. Collectively, the responses to these questions form the (raw) features stored in a 2-D table.There are 258 samples (each represented as a row) and 151 features (each represented as a column) in total. (See supplementary material for the survey questions.) After being preprocessed (detailed in Section 4.1), the table contains 209 features (plus the target variable for teaching the desired classification models): 2 continuous and 207 categorical.

Definition 1 (Skip Logic). In survey design, skip logic is a set of automated navigational rules that allows respondents to skip to relevant questions depending on their prior answers [Couper, 2008]. Each of the rules is defined as a *skip constraint*, which restricts the possible values of a subset of features A in accordance with the value of some feature a. We say the skip constraint on the *chain* A is *imposed* by a and denote this as $a \rightarrow A$. For the example in Figure 1, the skip constraint on $A = \{\text{TB4}\}$ is imposed by $a = \text{TB3}$ i.e., TB3 $\rightarrow \{\text{TB4}\}$, in which TB4 is *omissible* when TB3="No".

TB3 **In the past 6 months, have you smoked cigarettes?**
No (0) — SKIP TO TB5
Yes (1)

TB4 **How many cigarettes do you usually smoke in a day?**
Not at all (1)
Less than 1 cigarette a day (2)
1-5 cigarettes a day (3)
Half a pack a day (4)
A pack or more a day (5)

Figure 1: Skip Logic: If respondents answer "No" in TB3, they will automatically be directed to TB5 without being asked on TB4.

Definition 2 (Tabular Data Generation). Following the notations from [Xu *et al.*, 2019], given a table $\mathbf{T}_{real}$ that is partitioned row-wise into training set $\mathbf{T}_{train}$ and test set $\mathbf{T}_{test}$, the task involves training a data generator G on $\mathbf{T}_{train}$ and then independently sampling rows using the learned G to generate a synthetic table $\mathbf{T}_{syn}$ such that $|\mathbf{T}_{syn}| = |\mathbf{T}_{train}|$ with similar probability mass function for some target variable, y (defined in Section 2.2) (in order for us to fairly evaluate the efficacy of G).

Note that a capable G can satisfy the latter requirement without affecting the overall quality of the generated synthetic samples—that is, the features in each generated row should be consistent with the label in that row. For our setting, $\mathbf{T}_{real}$ and its derivatives consist of $N_c = 2$ continuous features and $N_d = 207 + 1$ categorical features.

Definition 3 (GAN and Its Extensions). GANs are *deep* generative models that have recently found success in modeling tabular data [Borisov *et al.*, 2022] in addition to images and text. A GAN typically consists of two separate networks: a generator $\mathcal{G}$ that maps a noise distribution (typically Gaussian) to the data distribution and a discriminator $\mathcal{D}$ that estimates the probability an input sample came from the data distribution. The learning process is defined as an adversarial game between $\mathcal{G}$ and $\mathcal{D}$ in which $\mathcal{G}$ attempts to consistently fool $\mathcal{D}$ [Goodfellow *et al.*, 2014].

To stabilize the training of GANs, [Arjovsky *et al.*, 2017] introduce WGAN that provides a meaningful loss, the Wasserstein distance, for quantifying the difference between the generated and real data distributions. Using the value function from WGAN-GP [Gulrajani *et al.*, 2017] (an improvement of WGAN), [Xu *et al.*, 2019] design CTGAN that aims to tackle class imbalance in categorical features of tabular data by modifying $\mathcal{G}$ to additionally take a vector as input. This so-called *conditional vector* represents a certain class/category of some categorical feature and is used to condition both the generated samples and the real training samples. The model can thus efficiently learn proper conditional distributions for each feature. We further describe the main components of CTGAN in Section 3.2.

2.2 Problem Formulation

In this work, we focus on *data augmentation* by designing a novel GAN to generate high-quality synthetic samples that resemble our tabular data with small sample size and skip logic incorporated. The augmented data is then used to train classifiers for predicting PWUDs' usage of a given drug within the next 12 months, including (A) whether they would increase its usage and (B) at which ordinal frequency they would use it. The former is a binary classification problem wherein PWUDs exhibiting an increase and non-increase (i.e., decrease or unchanged) in usage belong to the positive and negative class, respectively; the latter is a multiclass classification problem in which PWUDs are labeled according to their ordinal usage of the corresponding drug within 12 months after. We mainly concern with predicting usage of the most prevalent drugs (i.e., used by at least half of the PWUDs in our tabular data), which include marijuana, methamphetamine (or meth for brevity), amphetamines, and cocaine.

There are two unique properties of our tabularized survey data that impede existing generative and predictive models from achieving desirable performance: being high-dimension low-sample-size (HDLSS) and the presence of skip logic (whose impact is demonstrated in **Challenges and Observations** of the full paper).

3 Our Proposed GAN

3.1 Overview

We focus on GAN architectures due to their prevalence in tabular data generation literature [Borisov *et al.*, 2022] and their convenient properties (e.g., flexibility for conditional generation) that help us approach the discussed challenges systematically. Overall, we extend the popular CTGAN (reviewed in Section 3.2) as follows:

i) Due to the large number of (categorical) features, it is difficult for the conditional generator $\mathcal{G}$ to learn to generate samples that are conditioned on a particular feature. Therefore, during the training of $\mathcal{G}$, we raise the production of synthetic samples in order to ensure adequate training for conditional generation on a wide range of features. Furthermore, we prioritize the generation of synthetic samples that are conditioned on the empirical distributions of the target variable.

ii) To enhance the quality of synthetic samples *conditioned on* the target variable y, we incorporate an *auxiliary classifier* $\mathcal{C}$ [Park *et al.*, 2018] into the architecture of CTGAN for learning the correlation between y and other

features. That is, $\mathcal{C}$ is trained to predict the label of an input (synthetic or real) sample with y removed.

iii) Within $\mathcal{C}$, which is originally a multilayer perceptron (MLP), we add two auxiliary networks that together perform global feature selection and compute the weights of the first hidden layer of $\mathcal{C}$ [Margeloiu *et al.*, 2023]. The objectives of this addition are twofold: facilitate effective learning by $\mathcal{C}$ on HDLSS data and provide a mean to approximate the feature importance scores.

iv) Lastly, we enforce skip logic on synthetic samples during training by leveraging CTGAN's conditional vector.

3.2 Conditional Generator

We first revisit the two main components of CTGAN, the *conditional generator* and the *training-by-sampling* procedure. Formally, the conditional generator takes a vector $cond$ as additional input, which represents the condition ($D_{i^*} = k^*$) for some category k^* of the i^*th categorical feature $D_{i^*} \in \{D_1, \ldots, D_{N_d}\}$ (* denotes the selected feature/category in $cond$). It is worth noting that each condition only involves *one* category of *one* categorical feature and does not involve any continuous features. The sampling of $cond$ is twofold: sampling for the i^*th categorical feature and sampling for the k^*th category of that feature. Let $\oplus$ denote vector concatenation e.g., given $\mathbf{x}_1 = [0, 0, 0]$ and $\mathbf{x}_2 = [1, 0]$, $\mathbf{x}_1 \oplus \mathbf{x}_2 = [0, 0, 0, 1, 0]$. If each categorical feature D_i is encoded as a one-hot vector $\mathbf{d}_i = [\mathbf{d}_i^{(1)}, \ldots, \mathbf{d}_i^{(k)}, \ldots, \mathbf{d}_i^{(|D_i|)}]$, where $|D_i|$ is the number of categories in D_i and $\mathbf{d}_i^{(k)} \in \{0, 1\}$, then $cond$ is defined as $cond = \mathbf{m}_1 \oplus \ldots \oplus \mathbf{m}_{N_d}$ where $\mathbf{m}_i = [\mathbf{m}_i^{(1)}, \ldots, \mathbf{m}_i^{(k)}, \ldots, \mathbf{m}_i^{(|D_i|)}]$ is the *mask* vector of zeros associated with $\mathbf{d}_i$ with $\mathbf{m}_i^{(k)} = 1$ at $i = i^*$ and $k = k^*$. To ensure the generator produces samples in accordance with the given conditions, a cross-entropy term measuring the difference between $\mathbf{m}_{i^*}$ from the input $cond$ and the generated (denoted by $\hat{}$) one-hot feature $\hat{\mathbf{d}}_{i^*}$ is added to its loss.

To help the model evenly explore all possible categories in categorical features, a procedure for sampling the $cond$ vector, termed *training-by-sampling*, is employed in CTGAN as follows: randomly choose a categorical feature D_{i^*} with uniform probability; construct the probability mass function across the categories available in D_{i^*} by taking the logarithm of their frequencies in that feature (with respect to all training examples in $\mathbf{T}_{train}$); then sample a category k^* accordingly and calculate the corresponding $\mathbf{m}_{i^*}$ and $cond$. Afterward, the $cond$ vector is used to condition both synthetic and real training samples in order for the discriminator to properly estimate the (Wasserstein) distance between the learned and real conditional distributions $P_{\mathcal{G}}(\text{row}|cond)$ and $P(\text{row}|cond)$, respectively.

We design our GAN to place more emphasis on generating samples conditioned on y by leveraging the conditional generator (discussed in the following paragraph) as well as other known techniques (discussed in subsequent subsections).

Proper Conditional Generation in HDLSS Setting. In each iteration of CTGAN, $cond$ is sampled twice during the respective training of the discriminator $\mathcal{D}$ and the generator $\mathcal{G}$, with sample size equal the specified batch size for both. For our HDLSS setting with $|\mathbf{T}_{train}| < 200$ and over 600 columns/available conditions to consider, even when the batch size is maximally set to $|\mathbf{T}_{train}|$, any condition ($D_{i^*} = k^*$) would either be missing or inadequately sampled in the minibatch, and hence it would be impossible for $\mathcal{G}$ to properly learn to produce samples that preserve the input conditions. Therefore, we increase the sample size of $cond$ and hence the number of synthetic samples to generate during the training of $\mathcal{G}$ by a factor of $q > 1$ with respect to the batch size ($q = 1$ in CTGAN). Moreover, instead of randomly sampling D_{i^*} while constructing $cond$, we want $\mathcal{G}$ to sample certain features more frequently than others in order to prioritize learning their conditional distributions, particularly for the target variable y since our main purpose of generating synthetic samples is to help classification models improve their prediction on y. Therefore, we dedicate a portion of $cond$ to conditions solely for y. We discuss on how we sample the remaining D_{i^*}'s in Section 3.4.

3.3 Auxiliary Classifier

Incorporating an auxiliary classifier $\mathcal{C}$ into GAN architecture has been shown to improve conditional generation quality for both image and tabular data [Odena *et al.*, 2017; Park *et al.*, 2018]. $\mathcal{C}$ takes either a synthetic or a real sample having its label removed as input and aims to predict that label. Its loss, termed *classification loss* [Park *et al.*, 2018], quantifies the discrepancy between the label of a real sample and the label predicted by $\mathcal{C}$ for that same sample, which is formulated as binary and categorical cross entropy for Problems (A) and (B), respectively. The addition of $\mathcal{C}$ also introduces an extra loss term into the loss function of $\mathcal{G}$, which has the same form as the classification loss but concerns synthetic samples instead. We refer to this loss as *downstream loss* as in [Zhao *et al.*, 2022]. Altogether, $\mathcal{C}$ is trained to learn the actual correlation between the label and the features, then teach $\mathcal{G}$ how to generate realistic samples accordingly.

The synthetic samples that are fed to $\mathcal{C}$ are conditioned via the $cond$ vector that is sampled earlier during the training of $\mathcal{G}$ (we iteratively train $\mathcal{D} \blacktriangleright \mathcal{G} \blacktriangleright \mathcal{C}$). Since we are mainly concerned with learning the conditional distribution for the target variable $P_{\mathcal{G}}(\text{row}|y)$, we only feed synthetic samples that were conditioned on y to $\mathcal{C}$ when computing the downstream loss. The proper conditional generation step in Section 3.2 ensures that we have sufficient amount of such samples for training $\mathcal{C}$.

3.4 Learning Important Features

The auxiliary classifier $\mathcal{C}$ was originally proposed to be an MLP having the same architecture as $\mathcal{D}$ [Park *et al.*, 2018]. On HDLSS data, however, $\mathcal{C}$ is very likely to overfit, especially during the first few iterations and epochs when $\mathcal{C}$ encounters few training (either real or synthetic) examples, which in turn would negatively impact $\mathcal{G}$. Recently, [Margeloiu *et al.*, 2023] propose a way to overcome overfitting on HDLSS tabular classification problem by adding two auxiliary networks before the first hidden layer of some classification neural network in order to reduce the number of its learnable parameters and simultaneously perform global feature selection. We generalize this idea by integrating these networks into $\mathcal{C}$ to prevent it from overfitting. Fur-

thermore, by leveraging the global importance scores $\boldsymbol{s} = [s_1, \ldots, s_N] \in (0,1)^N$ (higher indicates greater importance) for N features, which are learned by one of the auxiliary networks, we can inform the conditional generator of important categorical features during training-by-sampling. More specifically, while sampling the $cond$ vector, we sample features D_i (excluding the target variable y) each with probability proportional to its important score s_i. Hence, conditions for features that have significant effects on the prediction of y are sampled more frequently, allowing $\mathcal{G}$ to prioritize generating synthetic samples conditioned on those features.

3.5 Enforcing Skip Logic

We first provide an intuitive explanation of how enforcing skip logic on synthetic samples can benefit the training of our GAN. Recall that CTGAN attempts to minimize the Wasserstein distance between $P_{\mathcal{G}}(\text{row}|cond)$ and $P(\text{row}|cond)$, where $cond$ represents some condition ($D_{i^*} = k^*$) as defined earlier in Section 3.2. Let $cond_1$ and $cond_2$ be two distinct samples of $cond$ with different sampled features e.g., D_1 ($i^* = 1$) and D_2 ($i^* = 2$), respectively (and hence different sampled categories). With the presence of skip logic, if D_1 and D_2 both belong to the same chain imposed by another feature D_3 and the corresponding skip constraint $D_3 \rightarrow \{D_1, D_2\}$ is enforced such that both D_1, D_2 are omissible, then $cond_1$ and $cond_2$ must be redefined to match the same conditional vector, $cond^*$, that satisfies such constraint i.e., ($D_1 =$ [BLANK]) AND ($D_2 =$ [BLANK]). It follows that on real data, $P(\text{row}|cond_1) = P(\text{row}|cond_2) = P(\text{row}|cond^*)$. This implies that the corresponding synthetic samples associated with either condition should follow the same distribution $P_{\mathcal{G}}(\text{row}|cond^*)$. Therefore, enforcing skip logic by inferring $cond^*$ effectively reduces the search space for $P_{\mathcal{G}}$, which leads to more efficient and stable learning and hence more consistency in the quality of the generated samples. We empirically demonstrate this claim in Section 4.2.

Existing methods for enforcing column-wise constraints require either creating customized transformation functions coupled with validity check or using reject sampling [Patki *et al.*, 2016], both of which are ad hoc and highly inefficient for large number of constrained columns. Instead, we leverage the $cond$ vector to enforce our constraints. Formally, recall from Section 3.2 that $cond$ is constructed as $\bigoplus_{i=1}^{N_d} \mathbf{m}_i$ with $\mathbf{m}_{i^*}^{(k^*)}$ set to 1 and all other entries set to 0 for representing the condition ($D_{i^*} = k^*$). Let us assume the corresponding categorical feature D_{i^*} from $cond$ imposes some skip constraint κ (e.g., $D_{i^*} =$ TB3 and $k^* =$ "No") on a chain of features $\{D_{i'}\}_{i' \in M}$, where $M \subseteq \{1, \ldots, N_d\}$ such that $|M|$ is the size of such chain. Let $k'_{i'} \in \{1, \ldots, |D_{i'}|\}$ be a specific category that each $D_{i'}$ takes in accordance with κ (e.g., $D_{i'} =$ TB4 and $k'_{i'} =$ [BLANK] under TB3="No"). Then, we can define κ as the extension of a condition as follows:

$$\begin{aligned} \kappa &= [D_{i^*} \rightarrow \{D_{i'}\}_{i' \in M}] \\ &= \left[(D_{i^*} = k^*) \Rightarrow \bigwedge_{i' \in M} (D_{i'} = k'_{i'}) \right]. \end{aligned} \tag{1}$$

Therefore, whenever applicable, we can restrict $cond \xrightarrow{\kappa} \zeta(cond)$ by reconstructing the individual mask vectors of $cond$ (with a slight abuse of notation) as

$$\mathbf{m}_i^{(k)} \xrightarrow{\kappa} \zeta\left(\mathbf{m}_i^{(k)}\right) = \begin{cases} 1 & \text{if } i \in \{i^*, i'\} \text{ and} \\ & k \in \{k^*, k'_{i'}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Note that κ is only defined for a fixed set of $\{k^*, k'_{i'}\}$ and hence can take various forms in practice. For instance, the following is an exhaustive list of valid expressions for $\kappa =$ TB3 $\rightarrow \{$TB4$\}$:

- TB3="No" $\Rightarrow$ TB4=[BLANK] (omissible)
- TB3="Yes" $\Rightarrow$ TB4="Not at all"
- TB3="Yes" $\Rightarrow$ TB4="Less than 1 cigarettes a day"
- TB3="Yes" $\Rightarrow$ TB4="1-5 cigarettes a day"
- TB3="Yes" $\Rightarrow$ TB4="Half a pack a day"
- TB3="Yes" $\Rightarrow$ TB4="A pack or more a day".

3.6 The Complete Model

The complete training procedure via minibatch stochastic gradient descent (SGD) is summarized in Algorithm 1. q was previously defined in Section 3.2. We denote the conditions for y as $cond^{(y)}$. $\omega \in [0,1]$ is the ratio for controlling the prevalence of such conditions in some sample of $cond$. Let $\mathcal{I}$ be the probability mass function across the categorical features (excluding y) wherein the probability for selecting a feature D_{i^*} is defined as its normalized feature importance score: $s_{i^*} / \sum_{i=1}^{N_d - 1} s_i$ (-1 for excluding y). The remaining conditions in $cond$ are sampled following the training-by-sampling procedure but with features selected according to $\mathcal{I}$ (the categories in each categorical feature are still sampled according to their log probabilities in $\mathbf{T}_{train}$ as before), for which we express as $cond_j \sim \mathcal{I}$ with a slight abuse of notation. The loss for $\mathcal{D}$, $L^{\mathcal{D}}$, is defined similarly as in CTGAN i.e., WGAN-GP loss. The loss function for $\mathcal{G}$ is $L^{\mathcal{G}} = L^{\mathcal{G}}_{orig} + L^{\mathcal{G}}_{dstream}$ where $L^{\mathcal{G}}_{orig}$ is the original loss function for $\mathcal{G}$ in CTGAN and $L^{\mathcal{G}}_{dstream}$ is the downstream loss defined in Section 3.3 along with the classification loss $L^{\mathcal{C}}$.

At each iteration of an epoch, the training sequence is as follows: train the discriminator (lines 3-6), train the generator (lines 7-10), and train the auxiliary classifier while simultaneously updating $\mathcal{I}$ according to $\boldsymbol{s}$ (lines 11-13).

4 Experiments

4.1 Methodology

Implementation Details

All experiments were conducted using PyTorch 1.13.1, CUDA 11.7, and scikit-learn 1.3.2. Our implementation of the proposed GAN[1] is based on CTGAN's. We refer readers to our full paper for further implementation details, particularly on our data preprocessing step.

Hyperparameters. For all considered generative models, unless otherwise stated, we adopt the same specifications as in the cited original work. We use a batch size of $|B| = 30$ to train each model, For CTGAN, we set the pac size [Lin *et al.*, 2018] to 3. $\mathcal{C}$ is a 2-layer MLP with (256, 256) neurons

[1] https://github.com/AnonyMouse3005/HDLSS-GAN

Algorithm 1 Training Our Proposed GAN

Input: Discriminator $\mathcal{D}$, generator $\mathcal{G}$, auxiliary classifier $\mathcal{C}$, constants q and ω

Initialize $\mathcal{I}$ uniformly
for each minibatch B of equal size from $\mathbf{T}_{train}$ **do**
  Sample noise Z and $cond$ (for training $\mathcal{D}$)
    $Z = [z_1, \ldots, z_{|B|}]$ with $z_j \sim \mathcal{N}(0, 1)$
    $cond = [cond_1, \ldots, cond_{(1-\omega)*|B|}]$ with $cond_j \sim \mathcal{I}$
  $cond \leftarrow \zeta(cond) \oplus [cond^{(y)}]_{\times \omega |B|}$
  Sample $|B|$ rows from $\mathbf{T}_{train}$ such that each row $\mathbf{r}_j$ satisfies $cond_j$
  Perform SGD update of $\mathcal{D}$ with $L^{\mathcal{D}}(\mathcal{G}(Z, cond), B)$
  Sample noise Z and $cond$ (for training $\mathcal{G}$ and $\mathcal{C}$)
    $Z = [z_1, \ldots, z_{q*|B|}]$ with $z_j \sim \mathcal{N}(0, 1)$
    $cond = [cond_1, \ldots, cond_{q*(1-\omega)*|B|}]$ with $cond_j \sim \mathcal{I}$
  $cond \leftarrow \zeta(cond) \oplus [cond^{(y)}]_{\times q\omega |B|}$
  Perform SGD update of $\mathcal{G}$ with $L^{\mathcal{G}}_{orig}(\mathcal{G}(Z, cond))$
  Perform SGD update of $\mathcal{G}$ with

$$L^{\mathcal{G}}_{dstream}\left(\mathcal{C}\left(\{\mathcal{G}(Z, cond_j)\}_{cond_j \in cond^{(y)}}\right)\right)$$

  Obtain $\boldsymbol{s}$ by computing $\mathcal{C}(B)$
  Update $\mathcal{I}$ according to $\boldsymbol{s}$
  Perform SGD update of $\mathcal{C}$ with $L^{\mathcal{C}}(\mathcal{C}(B))$
end for
Return: Trained $\mathcal{G}$

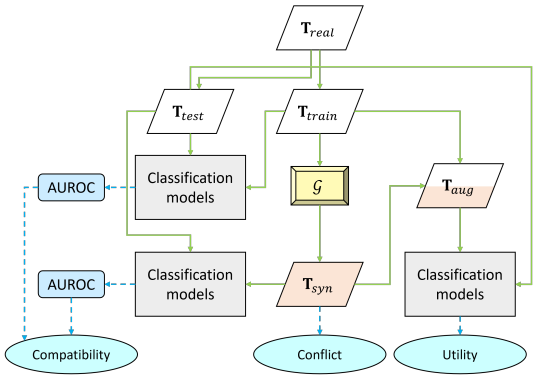


Figure 2: Workflow for evaluating the efficacy of generative models. The "Classification models" blocks refer to the same set of classifiers listed in the definition of Compatibility. Each of these blocks takes a table as input for training the classifiers and outputs their predictions on $\mathbf{T}_{test}$. The blue dashed arrows represent the computation of the respective scores/metrics.

and either a sigmoid (for Problem A) or a softmax (for Problem B) activation in the last layer. Each of the two attached auxiliary networks prior to the first hidden layer of $\mathcal{C}$ is a 4-layer MLP (256, 256, 256, 256) with a tanh and a sigmoid activation in the last layer, respectively. We manually tuned q and ω (via 10-fold cross validation on $\mathbf{T}_{train}$) before fixing their values to 20 and 0.5, respectively. Each model is trained for 100 epochs, each contains $\lfloor |\mathbf{T}_{train}|/|B| \rfloor$ iterations. The ratio for partitioning $\mathbf{T}_{real}$ is $80 : 20$ for $\mathbf{T}_{train}$ and $\mathbf{T}_{test}$, respectively, with a total of 100 distinct seeds.

Evaluation Metrics and Framework

Figure 2 illustrates our evaluation framework. We evaluate the efficacy of generative models[2] using three criteria: *conflict*, *compatibility* [Park *et al.*, 2018], and *utility*.

Conflict. Every row of a synthetic table generated by $\mathcal{G}$ should not contain too many entries that violate skip logic. Given the jth row of $\mathbf{T}_{syn}$ that is represented as $\hat{\mathbf{r}}_j = \hat{\mathbf{C}}_j \oplus \hat{\mathbf{d}}_{1,j} \oplus \ldots \oplus \hat{\mathbf{d}}_{N_d,j}$ where $\hat{\mathbf{d}}_{i,j}$ is the one-hot vector of the ith categorical feature and $\hat{\mathbf{C}}_j$ is the representation of continuous features in that row[3], we check for each skip constraint κ whether the columns of $\hat{\mathbf{r}}_j$ satisfy the condition $(D_{i^*} = k^*)$ i.e., match $cond$ (left-hand side of Equation 1). If it does, the one-hot vectors of the features within the chain linked by κ must exactly match the mask vectors associated with said features in the restricted $cond$ vector, $\zeta(\mathbf{m}_i)$ (whose construction is defined in Equation 2). Hence, we quantify the degree of κ-violation by computing the Hamming distance between the vectors $(\hat{\mathbf{r}}_{i,j})_{i \in M} = \bigoplus_{i \in M} \hat{\mathbf{d}}_{i,j}$ and $\bigoplus_{i \in M} \zeta(\mathbf{m}_i)$, where M contains the indices for the features within the chain linked by κ. The conflict metric of a single row $\hat{\mathbf{r}}_j$ is the average Hamming distance across all applicable skip constraints, and we compute the conflict of $\mathbf{T}_{syn}$ by taking its average across all rows. Thus, a synthetic table whose rows adequately conform skip logic yields a low conflict score in $[0, 1]$.

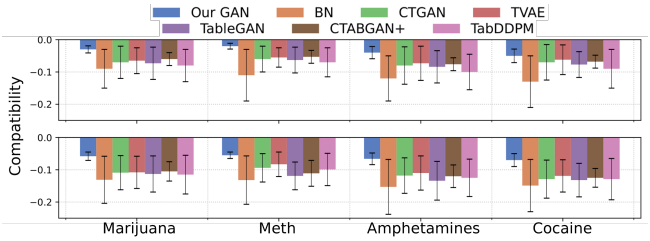
Compatibility. The classification models trained on synthetic data should output prediction for unseen examples in test data as accurately as those trained on real (training) data. We train classification models on $\mathbf{T}_{train}$ and on $\mathbf{T}_{syn}$ (both having the same size as defined in Section 2.1), then test them using $\mathbf{T}_{test}$ and compare their predictive performance, which is measured using the standard Area under the Receiver Operating Characteristic curve (AUROC). We train each of the multiclass classification models in Problem (B) using the one-vs-all strategy and we compute the resulting AUROC also using the one-vs-all strategy [Fawcett, 2006].

We report the average difference in AUROC (across different classification models [Xu *et al.*, 2019] and different partitioning of $\mathbf{T}_{train}$ and $\mathbf{T}_{test}$) of models trained on $\mathbf{T}_{syn}$ and those trained on $\mathbf{T}_{train}$. It is expected for the classification models trained on $\mathbf{T}_{syn}$ to score lower AUROC than those trained on $\mathbf{T}_{train}$, ideally with a margin as small as possible. Therefore, the compatibility score of $\mathbf{T}_{syn}$ should be negative and close to zero. We consider the following classifiers same for both Problems (A) and (B): elastic-net logistic regression [Zou and Hastie, 2005], decision tree i.e., CART [Breiman, 2001], random forest [Ho, 1995], XGBoost [Chen and Guestrin, 2016], CatBoost [Prokhorenkova *et al.*, 2018], 3-layer MLP (100, 100, 10) with sigmoid/softmax activation in the last layer, and WPFS [Margeloiu *et al.*, 2023].

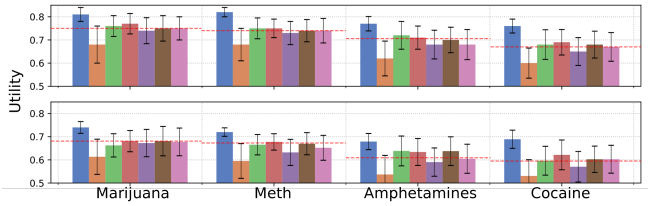
Utility. Ultimately, when the real training data is augmented by synthetic samples, the classification models trained on it should excel compared to the models trained on real data only. We report the average AUROC (across different classification models and different partitioning of $\mathbf{T}_{train}$ and $\mathbf{T}_{test}$) of the models trained on the augmented data, $\mathbf{T}_{train} + \mathbf{T}_{syn}$, and compare it against the average AUROC of those trained on $\mathbf{T}_{train}$. We consider the same set of

[2]For every measure of $\mathbf{T}_{syn}$'s quality while evaluating some trained generator G, we use G to generate 10 samples of $\mathbf{T}_{syn}$ (each satisfies the two requirements specified in Definition 2) and evaluate their quality independently, then average the respective scores.

[3]varied across different generative models e.g., CTGAN uses the proposed *mode-specific normalization* (and so does our GAN)



(a) Compatibility (greater i.e., less negative is better)



(b) Utility (greater is better)

Figure 3: Efficacy of considered generative models in Problem (A) (top) and Problem (B) (bottom). Each column considers one drug. The red dashed lines in Figure 3b mark the average AUROC of classification models trained on $\mathbf{T}_{train}$ without data augmentation.

classifiers listed above. Unlike compatibility, we do not compute the difference in AUROC scores. Hence, the utility of $\mathbf{T}_{syn}$ follows the same scale as AUROC $\in [0, 1]$ and should ideally yield values as high as possible.

4.2 Results

We use our evaluation framework to evaluate CTGAN, TVAE [Xu *et al.*, 2019], TableGAN [Park *et al.*, 2018], CTABGAN+ [Zhao *et al.*, 2022], and TabDDPM [Kotelnikov *et al.*, 2023] in addition to the proposed GAN, which are considered state-of-the-art for tabular data generation. For baseline, we use Bayesian networks [Pearl, 2011].

Compatibility of Synthetic Data. Figure 3a shows the performance of all considered generative models in terms of compatibility. Across all considered drugs in both problems, we see that the drop in predictive performance of classifications models trained on synthetic data that are generated from our GAN ($<$0.05 and $<$0.07 lower in AUROC for Problems A and B, respectively) is considerably smaller with lower uncertainty compared to the performance drop of those trained on synthetic data from other generative models. This demonstrates the efficacy of our GAN in generating synthetic samples that are comparable to real training data.

Utility of Synthetic Data. In terms of utility, the proposed GAN also performs well in each problem relative to the benchmarks as shown in Figure 3b. More specifically, the classification models trained using augmented data from our GAN gain from 8.35% up to 13.4% in AUROC in Problem (A) and 8.66% up to 15.8% in Problem (B), whereas those trained using augmented data from other models barely show any improvement. This implies that our GAN is capable of augmenting existing HDLSS data in order for classification models to effectively improve their predictive performance.

Violation of Skip Logic in Synthetic Data. The evaluation on degree of skip logic violation is summarized in Table 1. Since our work is the first to take this criteria into account, our proposed GAN has a clear advantage over existing models, with $\mathbf{T}_{syn}$ consistently having lower average conflict by a large margin in both considered problems. The additional runtime for enforcing all 26 skip constraints is negligible i.e., for approximately 5% longer training time.

Model	Problem (A)	Problem (B)
BN	0.471 ± 0.113	0.490 ± 0.154
CTGAN	0.339 ± 0.074	0.358 ± 0.085
TVAE	0.297 ± 0.072	0.325 ± 0.080
TableGAN	0.417 ± 0.108	0.452 ± 0.094
CTABGAN+	0.351 ± 0.086	0.379 ± 0.084
TabDDPM	0.377 ± 0.107	0.390 ± 0.090
Our GAN	$\mathbf{0.196 \pm 0.048}$	$\mathbf{0.218 \pm 0.039}$

Table 1: Conflict in $\mathbf{T}_{syn}$ (averaged over all splits of $\mathbf{T}_{real}$, lower is better). Both problems concern meth usage.

Impact of Enforcing Skip Logic. We also perform an ablation study to understand the practical benefits of enforcing skip logic during our GAN training. As shown in Table 2, when we enforce skip logic, the training phase exhibits not only higher stability but also higher efficiency, as evidenced by the loss of $\mathcal{G}$ at 50 and 100 epochs. Note that the adoption of WGAN (via CTGAN) in our GAN allows us to interpret the loss in a meaningful way. As a result, the trained GAN is able to consistently generate high-quality samples, which positively affects the scores for all three evaluation metrics to some extent. Similar results for other considered problems can be found in the appendix of our full paper.

Criteria	Enforcing Skip Logic?	
	No	Yes
$L^{\mathcal{G}}_{orig}$ @50 (↓)	-2.276 ± 0.849	$\mathbf{-3.472 \pm 0.653}$
$L^{\mathcal{G}}_{orig}$ @100 (↓)	-4.542 ± 0.933	$\mathbf{-4.871 \pm 0.795}$
Conflict (↓)	0.358 ± 0.096	$\mathbf{0.196 \pm 0.048}$
Compatibility (↑)	-0.026 ± 0.017	$\mathbf{-0.023 \pm 0.009}$
Utility (↑)	0.818 ± 0.038	$\mathbf{0.821 \pm 0.023}$

Table 2: Impact of enforcing skip logic during the training of our GAN in Problem (A) for meth. The (original) loss of $\mathcal{G}$ is recorded at 50 epochs ($L^{\mathcal{G}}_{orig}$ @50) and 100 epochs ($L^{\mathcal{G}}_{orig}$ @100). (↓) denotes less is better and (↑) denotes greater is better.

5 Conclusion

In this paper, using HDLSS tabular data collected by our team via a survey that employs skip logic on short-term substance use behavior, we design a novel GAN for augmenting our limited tabular data in order to help classification models accurately predict short-term substance use behaviors of PWUDs: (A) whether they would increase usage of a certain drug and (B) at which ordinal frequency they would use it within the next 12 months. Our evaluation results demonstrate the efficacy of the proposed GAN. The resulting predictions for the two defined problems can ultimately be leveraged by relevant substance use organizations as a complementary forecasting tool when determining the most appropriate resource to allocate to PWUDs that need the most help.

Acknowledgements

This project is supported by the National Institute of General Medical Sciences of the National Institutes of Health [P20GM130461], the Rural Drug Addiction Research Center at the University of Nebraska-Lincoln, and the National Science Foundation under grant IIS:RI #2302999. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

Ethics Statement

Due to the confidentiality agreement and IRB approval for this study, we do not have access to sensitive information such as full name and gender identity. Data will be made available to applicants (with an IRB protocol and ethical research plan) upon request.

References

[Andersson *et al.*, 2023] Helle Wessel Andersson, Mats P Mosti, and Trond Nordfjaern. Inpatients in substance use treatment with co-occurring psychiatric disorders: a prospective cohort study of characteristics and relapse predictors. *BMC psychiatry*, 23(1):1–10, 2023.

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[Borisov *et al.*, 2022] 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*, 2022.

[Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[Chawla *et al.*, 2002] 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.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[Colledge-Frisby *et al.*, 2023] Samantha Colledge-Frisby, Sophie Ottaviano, Paige Webb, Jason Grebely, Alice Wheeler, Evan B Cunningham, Behzad Hajarizadeh, Janni Leung, Amy Peacock, Peter Vickerman, et al. Global coverage of interventions to prevent and manage drug-related harms among people who inject drugs: a systematic review. *The Lancet Global Health*, 2023.

[Couper, 2008] Mick P Couper. *Designing effective Web surveys.* Cambridge University Press, 2008.

[Dillman *et al.*, 2014] Don A Dillman, Jolene D Smyth, and Leah Melani Christian. *Internet, phone, mail, and mixed-mode surveys: The tailored design method*. John Wiley & Sons, 2014.

[Fawcett, 2006] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[Fowler, 1995] Floyd J Fowler. *Improving survey questions: Design and evaluation*. Sage, 1995.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[Heckathorn, 2014] Douglas D. Heckathorn. Respondent-Driven Sampling: A New Approach to the Study of Hidden Populations*. *Social Problems*, 44(2):174–199, 07 2014.

[HHS, 2023] HHS. Sidebar: The Many Consequences of Alcohol and Drug Misuse — Surgeon General's Report on Alcohol, Drugs, and Health [Internet]. https://addiction.surgeongeneral.gov/sidebar-many-consequences-alcohol-and-drug-misuse, 2023. Cited: 2023-03-18.

[Ho, 1995] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[Karamouzian *et al.*, 2022] Mohammad Karamouzian, Andreas Pilarinos, Kanna Hayashi, Jane A Buxton, and Thomas Kerr. Latent patterns of polysubstance use among people who use opioids: A systematic review. *International Journal of Drug Policy*, 102:103584, 2022.

[Kotelnikov *et al.*, 2023] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[Lin *et al.*, 2018] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[Linden-Carmichael *et al.*, 2022] Ashley N Linden-Carmichael, Natalia Van Doren, Bethany C Bray, Kristina M Jackson, and Stephanie T Lanza. Stress and affect as daily risk factors for substance use patterns: An application of latent class analysis for daily diary data. *Prevention science*, 23(4):598–607, 2022.

[Liu *et al.*, 2022] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. Goggle: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2022.

[Lorvick *et al.*, 2023] Jennifer Lorvick, Jordana Hemberg, Madeleine J George, Joy Piontak, and Megan L Comfort. Protocol: Understanding polysubstance use at the daily and event levels: protocol for a mixed-methods qualitative and ecological momentary assessment study in a community-based sample of people who use illicit drugs in oakland, california, usa. *BMJ Open*, 13(9), 2023.

[Margeloiu *et al.*, 2023] Andrei Margeloiu, Nikola Simidjievski, Pietro Lio, and Mateja Jamnik. Weight predictor network with feature selection for small sample tabular biomedical data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9081–9089, 2023.

[Newcomb and Bentler, 1989] Michael D Newcomb and Peter M Bentler. Substance use and abuse among children and teenagers. *American psychologist*, 44(2):242, 1989.

[NIDA, 2020] NIDA. Introduction [Internet]. https://nida.nih.gov/research-topics/commonly-used-drugs-charts, 2020. Cited: 2023-03-18.

[NIDA, 2023] NIDA. Drug Overdose Death Rates [Internet]. https://nida.nih.gov/research-topics/trends-statistics/overdose-death-rates, 7 2023. Cited: 2023-07-11.

[Odena *et al.*, 2017] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[ODPHP, 2020] ODPHP. Drug and alcohol use. https://health.gov/healthypeople/objectives-and-data/browse-objectives/drug-and-alcohol-use, 2020.

[Ouimette *et al.*, 1997] Paige Crosby Ouimette, John W Finney, and Rudolf H Moos. Twelve-step and cognitive-behavioral treatment for substance abuse: A comparison of treatment effectiveness. *Journal of consulting and clinical psychology*, 65(2):230, 1997.

[Park *et al.*, 2018] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384*, 2018.

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

[Pearl, 2011] Judea Pearl. Bayesian networks. UCLA: Department of Statistics, 2011.

[Prokhorenkova *et al.*, 2018] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[Ray *et al.*, 2020] Lara A Ray, Lindsay R Meredith, Brian D Kiluk, Justin Walthers, Kathleen M Carroll, and Molly Magill. Combined pharmacotherapy and cognitive behavioral therapy for adults with alcohol or substance use disorders: a systematic review and meta-analysis. *JAMA network open*, 3(6):e208279–e208279, 2020.

[Russell *et al.*, 2021] Cayley Russell, Farihah Ali, Frishta Nafeh, Sean LeBlanc, Sameer Imtiaz, Tara Elton-Marshall, and Jürgen Rehm. A qualitative examination of substance use service needs among people who use drugs (pwud) with treatment and service experience in ontario, canada. *BMC Public Health*, 21(1):2021, 2021.

[SAMHSA, 2022] SAMHSA. Key Substance Use and Mental Health Indicators in the United States: Results from the 2021 National Survey on Drug Use and Health [Internet]. https://www.samhsa.gov/data/sites/default/files/reports/rpt39443/2021NSDUHFFRRev010323.pdf, 2022. Cited: 2023-07-11.

[Shorten and Khoshgoftaar, 2019] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[Street *et al.*, 1993] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. SPIE, 1993.

[Tanner-Smith *et al.*, 2016] Emily E Tanner-Smith, Katarzyna T Steinka-Fry, Heather H Kettrey, and Mark W Lipsey. *Adolescent substance use treatment effectiveness: A systematic review and meta-analysis*. Peabody Research Institute, Vanderbilt University Nashville, TN, 2016.

[SAMHSA, 2021] SAMHSA. Treatment episode data set: Admissions (teds-a). https://www.datafiles.samhsa.gov/dataset/treatment-episode-data-set-admissions-2021-teds-2021-ds0001, 2021.

[United Nations, 2023a] United Nations. Ensure healthy lives and promote well-being for all at all ages. https://sdgs.un.org/goals/goal3, 2023.

[United Nations, 2023b] United Nations. World drug report 2023. https://www.unodc.org/unodc/en/data-and-analysis/wdr-2023-online-segment.html, 2023.

[Wilson and Brown, 2023] Anna M Wilson and Aaron R Brown. Barriers to utilizing substance use disorder treatment and harm reduction services in appalachia. *Journal of Rural Mental Health*, 2023.

[Xu *et al.*, 2019] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

[Zhao *et al.*, 2022] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint arXiv:2204.00401*, 2022.

[Zou and Hastie, 2005] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.