IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 20 December 2023, accepted 18 January 2024, date of publication 22 January 2024, date of current version 29 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356913

RESEARCH ARTICLE

CasTGAN: Cascaded Generative Adversarial Network for Realistic Tabular Data Synthesis

ABDALLAH ALSHANTTI[1,2], **DAMIANO VARAGNOLO**[1], **ADIL RASHEED**[1], **ARIA RAHMATI**[3], **AND FRANK WESTAD**[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7034 Trondheim, Norway
[2]DNB ASA, 0131 Oslo, Norway
[3]Sopra Steria, 0185 Oslo, Norway

Corresponding author: Abdallah Alshantti (abdallah.a.s.alshantti@ntnu.no)

This work was supported by DNB ASA through the funding of this research project.

ABSTRACT Generative adversarial networks (GANs) have drawn considerable attention in recent years for their proven capability in generating synthetic data which can be utilised for multiple purposes. While GANs have demonstrated tremendous successes in producing synthetic data samples that replicate the dynamics of the original datasets, the validity of the synthetic data and the underlying privacy concerns represent major challenges which are not sufficiently addressed. In this work, we design a cascaded tabular GAN framework (CasTGAN) for generating realistic tabular data with a specific focus on the validity of the output. In this context, validity refers to the the dependency between features that can be found in the real data, but is typically misrepresented by traditional generative models. Our key idea entails that employing a cascaded architecture in which a dedicated generator samples each feature, the synthetic output becomes more representative of the real data. Our experimental results demonstrate that our model is capable of generating synthetic tabular data that can be used for fitting machine learning models, as CasTGAN's classification performance only falls under the real training data's PR-AUC score by 4.88% on average for classification datasets, and exhibits an average reduction of the real training data's R^2 score by 0.139 for regression datasets. In addition, our model captures well the constraints and the correlations between the features of the real data, especially the high dimensional datasets. Assessing the generation of invalid records demonstrates that CasTGAN reduces the number of invalid data observations by up to 622% in comparison to the second best performing baseline tabular GAN model. Furthermore, we evaluate the risk of white-box privacy attacks on our model and subsequently show that applying some perturbations to the auxiliary learners in CasTGAN increases the overall robustness of our model against targeted attacks.

INDEX TERMS Generative adversarial networks, output validity, privacy attacks, tabular data.

I. INTRODUCTION

Facilitating information and knowledge sharing within and between organisations is increasingly sought after for attaining growth and development. From a healthcare and medical standpoint, information exchange subsequently contributes to better understanding of diseases and risk factors, more intuitive prognosis by practitioners and effective treatment planning based on previously obtained knowledge [1]. In the financial sector, sharing information between stakeholders leads to improved prediction of corporate bankruptcy and quicker identification of suspicious transaction behaviour that can be potentially linked to organised financial crime [2]. For both fields, sharing the data which contains sensitive patient and client information is subject to the European Union's General Data Protection Regulation (GDPR) [3] to

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

maintain the confidentiality and privacy of such information. Therefore, institutions are continuously seeking new data anonymisation and synthetic data generation techniques for exchanging domain knowledge without exposing sensitive information.

Ever since their development, generative adversarial networks (GANs) [4] have been increasingly studied for their ability to approximate and model complex data distributions. Despite the early GAN applications being densely focused on the computer vision domain and image generation, GANs are becoming recently researched in other fields such as natural language processing [5] and time-series anomaly detection [6]. In addition, more properties of GANs have emerged such as conditionally generating samples based on a specific target class [7] and generation in conjunction with variational auto-encoders [8].

In contrast, GANs have been significantly less explored for tabular data generation. A tabular dataset typically comprises a mixture of continuous variables and categorical features [9]. Tabular data is common in the medical and the financial domains where fields such age, gender, profession, and income can be commonly found in databases containing numerous records. As opposed to purely numerical data, representing datasets with categorical variables can be particularly difficult in presence of highly-dimensional and strongly correlated features. Furthermore, quantifying the validity of a synthetic tabular dataset can be practically impossible without closely inspecting every generated data sample and deciding whether to accept or reject each examined data record. Correspondingly, invalidity refers to the semantically incorrect representation of the features, where the interdependencies between some data features are not correctly modelled. This constitutes a challenge when synthetic data is harnessed for knowledge exchange, as semantically incorrect data can lead to misinterpretation and flawed understanding of the data, hence disparaging the effect of facilitating knowledge sharing. Notwithstanding, there currently exists no straightforward and unified criteria for evaluating the validity of the output generated by tabular GANs [10].

To rectify the previously outlined limitations, we introduce CasTGAN, which is a generative network framework characterised by multiple generators connected sequentially; each of which is designed to generate a single feature. Meanwhile, a single discriminator validates the output of all the generators while being trained on the output of the final generator in the cascade. In addition, each generator is chained to a corresponding auxiliary learner in order to obtain more insightful losses specific to the individually generated features. This is motivated by the fact that it has been shown that adding more auxiliary classifiers can enhance the quality of the synthetic output images [11]. Therefore, we posit that CasTGAN aims to capture the highly correlated and hierarchical relationship between features, such that the synthetic output produced by our model closely resembles the real data while minimising the inconsistencies in the generated data. This is particularly important for applications where data is widely shared between professionals, and the slightest irregularities in the data can lead to undesired outcomes.

We can thereby summarise our contributions in this work as:

- *Generative architecture*: A cascaded based generative framework for producing realistic tabular output which greatly emulates the original data, while significantly reducing the number of invalid synthetic samples.
- *Synthetic data evaluation*: A new metric for quantifying the *realistic-ness* of the synthetic data when lacking the domain knowledge for the provided data, and extensively evaluate our framework and existing works.
- *Privacy assessment*: We launch white-box privacy attacks on our model and analyse how the privacy guarantees and quality of the output are impacted when perturbing the input data during the model training.

The remainder of this paper is structured as follows. In Section II, we present an overview of GANs and the types of GAN privacy attacks, while we further examine the relevant studies in Section III. Section IV presents a discussion of CasTGAN and a detailed description of our model's structure. In Section V, we outline the experimental setup used in this work and the evaluation criteria. We demonstrate our results in Section VI and discuss our findings in Section VII. The paper is concluded in Section VIII.

II. BACKGROUND

A. GENERATIVE ADVERSARIAL NETWORKS

A GAN is characterised by a generator G and a discriminator D playing an adversarial game, where each component attempts to maximise its own benefit [4]. The generator receives a noise input sampled from a random distribution $z \sim p_z$ and learns to generate an output in the distribution $x \sim p_g$ that matches the structure of the unseen real data $x \sim p_{data}$. Meanwhile, the discriminator has access to the samples produced by the generator and the real data, and learns to distinguish between its real and fake inputs. While the output generated by G improves during training as a result of the loss it obtains from the discriminator, the discriminator also becomes increasingly clever in recognising the data produced by the generator. Subsequently, GANs are particularly challenging to train since it must be guaranteed that both the generator and the discriminator maintain their competitiveness without outperforming each other early in the training phase. In the classic GANs, the generator and the discriminator attempt to maximise their objective by minimising the Jenson-Shannon Divergence (JSD), however, using JSD does not guarantee the convergence of losses, hence leading to training instability [4].

The Wasserstein GAN (WGAN) has been proposed as an alternative to the standard GANs in order to augment the training stability in generative models by replacing JSD with the Wasserstein Distance (WD) [12]. The use of the Wasserstein Distance ensures that the model is continuously learning even if the quality of the output is poor, and this is attributed to the smooth gradients produced by the Wasserstein cost function. The initial WGAN relied on weight clipping to enforce the confinement of the discriminator's weights within a specified range $[-c, c]$, however, the authors demonstrate that clipping can lead to difficulties with model optimisation [12]. Instead, the use of gradient penalty with WGAN has been proposed to mitigate against the exploding and vanishing gradients of the weights [13]. In this setting, gradients are found using the linear interpolations $\hat{x} \sim p_{\hat{x}}$ between the real and the fake samples, where the distribution of linear interpolation is resembled by $p_{\hat{x}}$. Additionally, the gradient penalty coefficient λ_{GP} is used as a parameter for controlling the level at which the gradient penalty affects the discriminator.

The objective function for the WGAN-GP can therefore be represented as:

$$\underset{G}{min}\,\underset{D}{max}\; V(D)\, \mathbb{E}_{x\sim p_{data}}\left[D(x)\right] - \mathbb{E}_{z\sim p_z}\left[D\left(G(z)\right)\right] \\ - \lambda_{GP}\, \mathbb{E}_{\widehat{x}\sim p_{\hat{x}}}\left[(\|\nabla_{\widehat{x}} D\left(\widehat{x}\right)\|_2 - 1)^2\right] \quad (1)$$

WGAN-GP is increasingly becoming more prevalent than the classic GANs in applications such as image generation and tabular data generation, as it contributes to more stable learning. In addition, WGAN-GP minimises the effect of mode collapse - that is when the generator learns to "trick" the discriminator by producing a limited number of modes which the discriminator incorrectly classifies as real samples, instead of utilising the entire data feature space.

B. PRIVACY ATTACKS

In machine learning, membership inference attacks (MIA) aim to identify whether a data sample was used in the training of a machine learning model [14]. For instance, the attackers might try to identify whether the records belonging a client were used for training a loan default prediction model. In this case, the attackers' objective would be to determine whether the client has taken a bank loan, with such information being used for targeted fraud attempts. Privacy guarantees in machine learning have been extensively studied in the form of analysing the connection to model overfitting [15] and in differential privacy [16].

More recently, MIA have also been explored for generative models. Privacy attacks applied on synthetic samples generated by GAN models aim to reconstruct the real data samples which were used in GAN training. In principle, membership attacks on GANs can be categorised into three types of attacks [17]; *full black-box* in which attackers have access to only synthetic samples, *partial black-box* where the attackers have access to the synthetic samples and the latent codes used to generate them, and *white-box* which assumes that the attackers are able to access the internal parameters of the generator, the discriminator or both. The trade-off between the quality of synthetic samples and the privacy guarantees of GANs have been additionally examined in existing works [18], [19], [20].

III. RELATED WORKS

Tabular data is broadly used in regression and classification tasks, which facilitates a growing interest in tabular data synthesis for machine learning applications, especially in domains with limited training data. Bayesian networks can be used for generating synthetic records by approximating the conditional probability distribution from the data [21]. While the Bayesian networks can in practice be additionally used for exploring causal relationships between the independent variables, estimating the distributions is often built on simplifying assumptions on the data [22]. Meanwhile, tree-based methods were first utilised for generating partial synthetic data in [23], and has further explored in [24] where adversarial random forest has demonstrated comparable performance to deep learning techniques in terms of synthetic data quality. However, the privacy-utility trade-offs for synthetic data generation using tree-based density estimators is not sufficiently explored.

Deep neural networks have been widely studied for synthetic data generation, thanks to their capabilities for handling and approximating the distributions of large datasets. Variational auto-encoders (VAEs) [25] estimate the probabilistic distribution of by finding a lower-dimensional latent representation of the data. The application of VAEs has been extended to image data generation [26], oversampling of anomaly event data [27] and tabular data synthesis [28]. An underlying limitation with variational autoencoders is the assumption of the a simple parametric form of the latent space, which leads to a difficulty in capturing complex data distributions [29]. Invertible neural networks have also been proposed for tabular data synthesis through variants based on neural ordinary differential equations [30], copula flows [31] and normalizing flows for private tabular data generation [32]. A drawback of the invertible neural network based synthesis is that extensive hyperparameter tuning is needed to achieve satisfactory classification and regression performance.

Within deep learning based generative models, GANs are favourable due to their ability to generate complex synthetic data in an adversarial and unsupervised setting [37]. table-GAN [33] is one of the earliest generative models for producing tabular synthetic output based on adversarial training. Using convolutional neural networks, table-GAN demonstrates that GANs unsurprisingly outperform anonymisation techniques while highlighting the potential privacy risks arising from membership attacks. Meanwhile, in medGAN [34] an autoencoder based generative model is developed for generating high-dimensional medical patient records while shedding light on the privacy risks attributed to the generated data. A long-short term memory (LSTM) architecture for the generator was adopted in [38] demonstrating the potential of recurrent neural networks in synthetic

TABLE 1. Properties of some state-of-the-art tabular GANs, used as baselines later Section VI along with our proposed model, CasTGAN. When counting the number of datasets, we only consider real datasets as opposed to simulated datasets with known distributions and priors. While some methods conduct privacy analysis by measuring distance to closest records, this is notably different than evaluating the threat posed by privacy attacks.

Method	Data types	No. of real datasets	Machine learning utility evaluation	Univariate similarity evaluation	Correlation analysis	Validity evaluation	Privacy attacks evaluation
table-GAN [33]	mixed	4	✓	✓	✗	✗	✓
medGAN [34]	continuous	3	✓	✓	✗	✗	✓
CTGAN [28]	mixed	6	✓	✓	✗	✗	✗
cWGAN [35]	mixed	7	✓	✓	✗	✗	✗
CTAB-GAN [36]	mixed	5	✓	✓	✓	✗	✗
CasTGAN [ours]	mixed	6	✓	✓	✓	✓	✓

data generation. A GAN approach based on an autoencoder and Kullback-Liebler divergence to tackle mode-collapse was proposed in [39], in which high-quality synthetic output was produced, with comparable performance to state-of-the-art methods. In CTGAN [28], conditional training of a GAN is carried out by instructing the generator and the discriminator to sample based on randomly selecting feature category in every training iteration, where a highly realistic tabular output can be observed from their evaluation. In [35], the authors propose cWGAN, which is a GAN-based oversampling technique focusing on the generation of samples belonging to the minority class in financial credit datasets. Zhao et al. [36] builds up on existing tabular GAN models by employing convolutional neural network (CNN) and conditional vectors to improve the representation of skewed distribution of numerical features of the synthetic output. Finally, Strelcenia et al. [40] comprehensively review the existing tabular GAN literature and highlight that an underlying limitation of the tabular GAN-based studies is the lack of standarised evaluation metrics.

Acknowledging the aforementioned, it is manifested that there is no shortage of novelties in synthetic tabular data generation literature. Nevertheless, an underlying challenge remains the proposal of evaluation techniques and criteria for quantifying the reliability and the statistical properties of the synthetic data. A further limitation is the sufficient analysis of data with hierarchical and interdependent variables. Therefore, our focus in this work is proposing a new framework that alleviates the two preceding deficiencies in the synthetic tabular data domain. A general overview of the properties, strengths and limitations of our model and the existing studies that we adopt as baselines is outlined in Table 1.

IV. METHODOLOGY

Generating synthetic data from unknown and correlated distributions is a non-trivial task. The architecture of CasTGAN is tailored for generating mixed-type features that have similar distributions to the ones observed in a real dataset. Additionally, cascaded structures enable modelling correlations among features in a sequential manner. Given a dataset X with M features, the features of the dataset can be represented as $\{m_1, m_2, \ldots, m_M\}$.

A. MODEL ARCHITECTURE

The proposed CasTGAN framework is characterised by a cascade $\vec{G}$ of multiple generators, $\vec{G} = \{G_1, G_2, \ldots, G_M\}$, in which generators $G_1, G_2, \ldots, G_M$ are connected sequentially and coupled with auxiliary learners $AL_1, AL_2, \ldots, AL_M$. Generator G_i and auxiliary learner AL_i for $i = 1, 2, \ldots, M$ are devoted for feature m_i in the dataset, and the real data is used for fitting the auxiliary learners and the discriminator D. An illustration of the CasTGAN architecture is depicted in Figure 1.

As can be visualised from Figure 1, each generator G_i focuses on generating its target feature using a primary neural network. The cascade of generators are laid out sequentially such that generator G_i obtains its inputs from a given noise vector z whose components are standard Gaussian and i.i.d., and from the outputs of the previous generator – the only exception being the first generator which only takes a vector of random noise as its input.

Notation-wise, generator G_i takes as input ϕ_i two objects: the useful outputs coming from G_{i-1} (i.e., the vector $\check{G}_{i-1}(\phi_{i-1})$) and the noise vector z (note that the same vector is fed to all the generators as depicted in Figure 1). The generator G_i then produces one output, i.e., $G_i(\phi_i)$, that may though be logically split in three distinct components: Z_i, that will be considered redundant information and that will not be used by the next generator; $\widehat{X}_i$, that is the target feature of generator i; and $\check{G}_{i-1}(\phi_{i-1})$, that is simply the information from the past generator that will be forwarded to the next one. The input and the output of generator G_i are shown in Figure 2.

Formally, the output of generator G_i can be presented as:

$$G_i(\phi_i) = \check{G}_i(\phi_i) \oplus Z_i, \tag{2}$$

while the information that generator G_i will pass to G_{i+1} is

$$\check{G}_i(\phi_i) = \begin{cases} \widehat{X}_i & \text{if } i = 1 \\ \widehat{X}_i \oplus \check{G}_{i-1}(\phi_{i-1}) & \text{if } i \geq 2\ . \end{cases} \tag{3}$$

Note that the generator is actually composed by two distinct neural networks: the primary one, whose input is

$$\phi_i = \begin{cases} z & \text{if } i = 1 \\ z \oplus \check{G}_{i-1}(\phi_{i-1}) & \text{if } i \geq 2, \end{cases} \tag{4}$$

and the secondary neural network, whose input is the noise vector z above and whose output Z_i is the redundant information output mentioned above, that will not be passed forward to G_{i+1} but will instead be used by AL_i.

We note that, as the losses are not backpropagated to the secondary neural network, G_i retains its primary objective of generating its target feature based on the input provided to it.

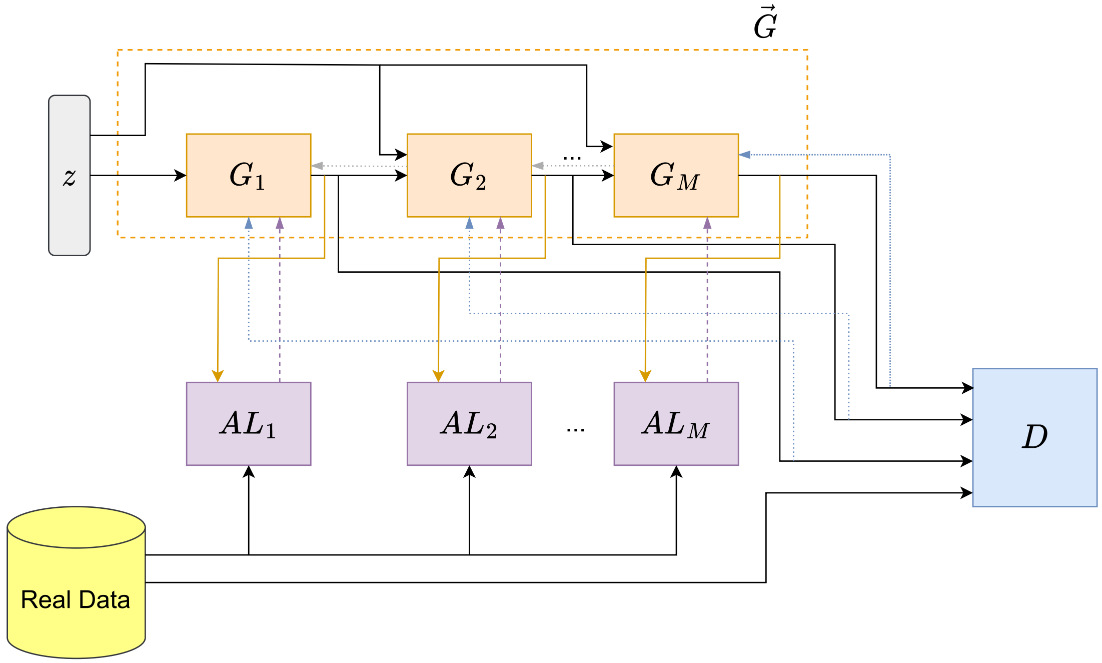


FIGURE 1. The model architecture of CasTGAN. The cascade $\vec{G}$ is composed of generators $G_1, G_2, \ldots, G_M$ sequentially lined up. The auxiliary learners $AL_1, AL_2, \ldots, AL_M$ are fitted on the real data, and are utilised by their respective generators for querying the generation of the data features $m_1, m_2, \ldots, m_M$. The cascade of generators $\vec{G}$ takes noise vector z as input, while the discriminator D is trained to distinguish the real data from the synthetic data. As depicted, generators $G_1, G_2, \ldots, G_{M-1}$ receive three losses: loss directly from the discriminator, loss backpropagated from the previous generator and the loss from the auxiliary learner. Meanwhile, G_M is passed the loss from its auxiliary learner and the loss from the discriminator.

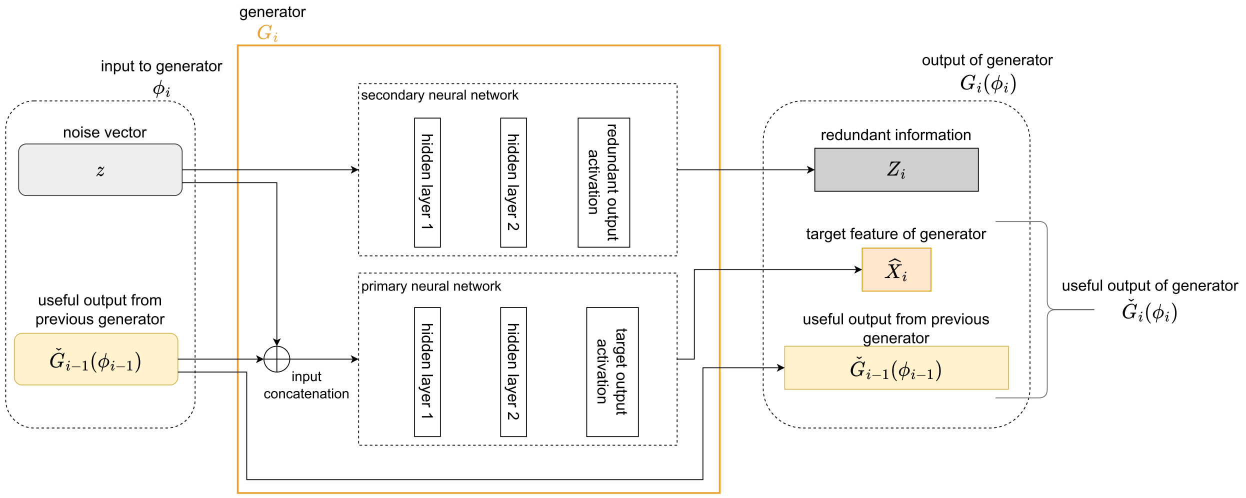


FIGURE 2. A close in visualization of generator G_i in our GAN architecture. The structure in the figure is applicable to all generators in the cascaded layout, except for G_1, which receives only noise vector z as input.

Summarizing, the overall cascaded generator structure can be denoted as

$$\vec{G}(z) = \widehat{X}, \tag{5}$$

where $\widehat{X}$ is the generated synthetic output.

Based on our literature survey, we observe that some of state-of-the-art tabular GANs employ a conditional setting to enforce the representation of features in both the generator and discriminator [28], [36]. We note that while this is indeed an effective strategy for representing discrete categories and preventing mode collapse, this approach is considerably inefficient for sampling datasets with a large number of categories and few data records, since conditioning on a single random category at every training iteration might not be sufficient to cover all the existing categories in the dataset. In this paper we seek to analyse whether, how much and under which conditions resorting to a series of auxiliary learners – one for each feature – may encourage the models to learn to represent based on the losses traversed, rather than explicitly constraining the model output. The hypothesis is indeed that if multiple auxiliary losses are computed in parallel, the model might be able to improve the learning of categorical interdependence and scale up accordingly to highly dimensional tabular datasets.

B. AUXILIARY LEARNERS ARCHITECTURE

GANs for tabular data synthesis are known to be prone to training instability and mode collapse due to the imbalanced feature categories [41]. Conditional GANs [7] have been deployed to generate synthetic output belonging to specific classes. Conditioning both on the generator and the discriminator has been shown to stabilise the training process of GANs.

On the other hand, the use of auxiliary learning for predicting the target variable given the data represents an alternative approach for capturing the characteristics of the data attributed to given target feature. It has been demonstrated that the auxiliary loss further stabilises the training process in comparison to conditional generation, and leads to a representation that is independent of target label [42]. We observe that while auxiliary learners are traditionally embedded within the discriminator [11], we instead propose designing auxiliary learners as independent structures.

In the CasTGAN, we craft M auxiliary learners $AL_1, \ldots, AL_M$ for learning to predict the individual features. Due to its scalability on large datasets and the relatively fast convergence speed, we focus on building the auxiliary learners using the Light Gradient Boosting Machine (LightGBM) [43], which we pre-train prior to the GAN training. An auxiliary learner AL_i corresponding to feature m_i is trained on $X_{\notin i}$ in order to predict X_i. Following standard strategies for such tasks, for predicting the numerical features, the mean-squared error loss is used in the training of the auxiliary learners, whereas cross-entropy loss is used for predicting the categorical and binary variables. As with other decision tree based models, there is no need to one-hot encode the categorical features in $X_{\notin i}$, but instead the categories are converted into integer encodings. Meanwhile the LightGBM auxiliary learners are capable of handling numerical features with extreme magnitudes, and therefore numerical features are not scaled for auxiliary training.

As LightGBM models have low computational complexity and are generally fast to to train [43], assigning an auxiliary learner for every feature is a reasonable approach for representing the auxiliary loss L_{AL} for predicting a feature given all the other features. It is worth noting that for the early auxiliary learners in the cascaded sequence AL_1 to $AL_{\lceil M/2 \rceil}$, the generated data feature space $\widehat{X}_{\notin i}$ is heavily dominated by redundant variables Z which subsequently lead to increased auxiliary losses. However, these losses help the early generators in producing features that closely match the distributions of the training data. Meanwhile, the task for the later generators and auxiliary learners in the cascade becomes increasingly focused towards generating features that can be predicted from the initially generated target features $\check{G}_{i-1}(\phi_{i-1})$.

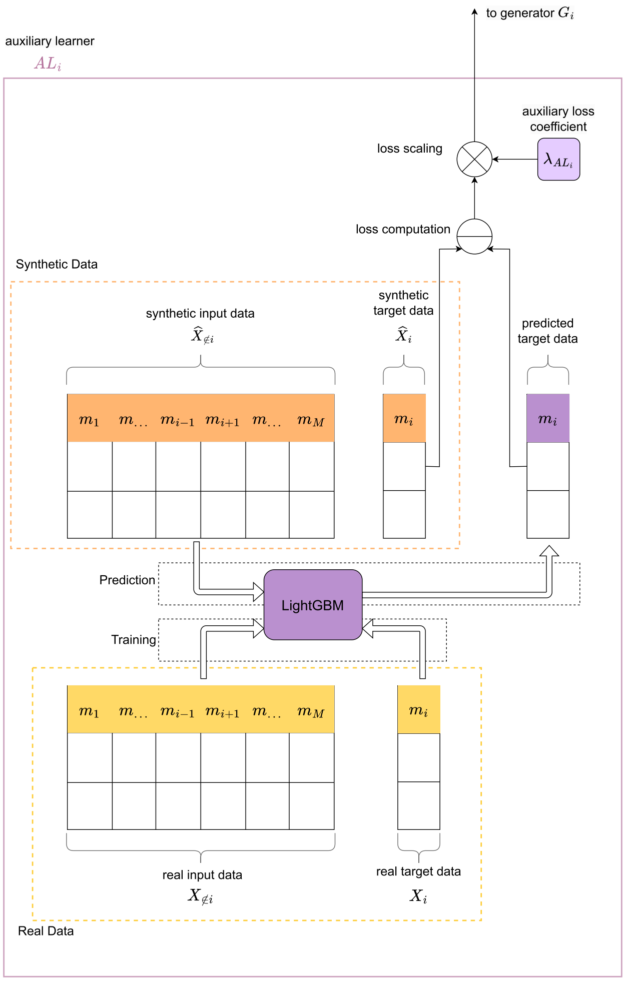


FIGURE 3. A detailed depiction of auxiliary learner AL_i in CasTGAN. The training of the LightGBM model on the real data occurs prior to the training of the GAN model. Meanwhile, the synthetic data from the generator G_i is queried against the auxiliary learner AL_i during the training iterations of the GAN to compute the auxiliary loss of the generator's target feature.

To ensure that the auxiliary losses $L_{AL_1}, \ldots, L_{AL_M}$ do not overexceed the generator's ones, there is the need for scaling down the losses from the auxiliary learners. In this paper we analyze the choice of performing this scaling down by means of constant coefficients λ_{AL}. In principle, λ_{AL} could be a single scalar value that is applied to all the auxiliary losses. However, we consider a vector of auxiliary loss coefficients $\lambda_{AL_1}, \ldots, \lambda_{AL_M}$ since it is fundamentally important for the early generators to generate variables that conform to the original feature distributions, since this will prompt the next generators in the cascade to effectively learn the feature correlations. Though this a hyperparameter, we set $\lambda_{AL_1} = 0.75$ and $\lambda_{AL_M} = 0.10$ while the auxiliary coefficients in between are linearly and equidistantly scaled in the $[0.75, 0.10]$ range in our experiments. The same auxiliary setting is applied for all the datasets in this work. The overall structure of auxiliary learner AL_i is illustrated in Figure 3.

C. TRAINING DATA TRANSFORMATION

The main novelty in this paper comes from testing the effects of designing multiple generators in a cascaded layout, where each generator focuses on generating a single data feature. To represent the numerical features we propose to use Variational Gaussian Mixture models (VGM) [44] to estimate the number of modes for a numerical feature, as it has been demonstrated that correctly representing multi-modal numerical data objectively reduces the incidence of mode collapse [39]. In this context, each mode is essentially a Gaussian model on its own, where in the transformation process a single mode is selected for a feature and a scalar value is calculated for quantifying the magnitude of the mode. As such, the transformation of a numerical feature changes the initial unscaled real number into a vector of size equivalent to the number of Gaussian mixture models + 1 (the *1* being the magnitude value of the respective mode and the vector being a one-hot encoded representation of the selected mode). The representation of continuous features using variational Gaussian models is not exclusive to this work as it has been adopted with notable success in earlier tabular GANs [28], [36].

Meanwhile, the categorical features of the training data are transformed into one-hot encodings before being fed to the discriminator. For GANs, the one-hot encoding vectorisation of the categorical features presents an intuitive approach to process the data as it can be scaled and can be appropriately used by the model without issues such as exploding gradients. Furthermore, the use of one-hot encoding simplifies the task of introducing non-linearities by the generator for guaranteeing that the model gradients are differentiable. It is worth reiterating that categorical and numerical transformations of the data for use by the generator and discriminator differ from the representation of the same data used for training and evaluating the auxiliary learners.

D. GENERATORS AND DISCRIMINATOR

The generators receive input in the form of noise vector z and the untransformed meaningful output $\check{G}_{i-1}(\phi_{i-1})$. As with other GAN applications, we highlight that using a larger noise vector leads to a better output of the features and can mitigate against mode collapse [39]. Throughout all the experiments, we use a noise vector of size 128, though this is a hyperparameter that can be tuned accordingly [45]. Since each generator dedicates its effort into generating one feature at a time, we use a simple primary neural network of hidden sizes (128, 64). Additionally, we use layer normalisation after the hidden dimension [46] for standardizing the weights into zero mean and unit variance and for speeding up the training process. We also use the LeakyReLU activation function with a small negative slope as opposed to ReLU in order to remove the constraints associated with setting the negative gradients to zero.

The dimension of the output layer of the generator for producing the target feature is equivalent to the number of one-hot encodings if the feature is categorical or equal to the number of VGM modes +1 if the target feature is numerical. A hyperbolic tangent (`tanh`) activation is applied to the scalar value of the numerical VGM representation. For the categorical output and the one-hot encoded vector of the VGM vector we use gumbel softmax activations for introducing non-linearities to the output. The Gumbel-softmax works by adding noise from the Gumbel distribution to the vectorised logit output of the generator while maintaining the differentiable nature of the GAN training. The Gumbel-softmax function exhibits also a temperature parameter τ that may be used to control the diversity of the output generated by the function. In our experiments $\tau = 0.8$ was assessed as proper to generate a diversified output that reduces the effects of mode collapse, while conforming to the distribution of categories of the feature within the training set.

We then note the risk that the discriminator may learn to distinguish between real and generated data by discriminating between the hard one-hot encoded real data and the float values from the generator. To minimize this risk we add an i.i.d. Gaussian noise distributed as $\mathcal{N}(0, 0.01)$ [12] to the columns of the real samples before feeding them to the discriminator. Consequently, all the inputs to the discriminator (i.e., numerical and categorical features of real and generated data) are float values. The weights of the discriminator are then trained using only the outputs from the final generator G_M. The parameters of the various generators are though updated based on the loss of the discriminator, that is thus computed for this reason.

We maintain a simple architecture for the discriminator comprising two hidden layers of sizes (256, 128). As with the generator, layer normalization and LeakyReLU activations are used between the hidden layers. The final layer consists of a single output node without an activation function. To alleviate against mode collapse & GAN training instability issues we additionally compute the Wasserstein loss [12] with gradient-penalty [13] for the calculation of the discriminator losses.

As the CasTGAN is built up using M generators, we have multiple min-max games between the generators and the discriminator. Therefore, the value function for the discriminator can be expressed as

$$\begin{aligned}\min_{\vec{G}} \max_{D} V(D) = \quad & \mathbb{E}_{x \sim p_{data}} [D(x)] \\ & - \mathbb{E}_{z \sim p_z} \left[D\left(\vec{G}(z) \right) \right] \\ & - \lambda_{GP} \mathbb{E}_{\widehat{x} \sim p_{\widehat{x}}} \left[(\| \nabla_{\widehat{x}} D\left(\widehat{x}\right) \|_2 - 1)^2 \right] \end{aligned} \tag{6}$$

and value function for generator G_i is hence given by

$$\begin{aligned}\min_{G_i} \max_{D} V(G_i) = & - \mathbb{E}_{\phi \sim p_\phi} \left[D\big(G_i(\phi)\big) \right] \\ & + \lambda_{AL_i} \mathbb{E}_{\phi \sim p_\phi} \left[L_{AL_i} \right]. \end{aligned} \tag{7}$$

V. EXPERIMENTAL SETUP

Evaluating the performance of GANs is a non-trivial task, and this is evident from literature, where no standard approach for evaluating GANs on tabular data can be found. In this section, we describe the experimental design for CasTGAN, with a thorough discussion of the metrics used for the model's analysis. We implemented our model using PyTorch in Python on a Linux Ubuntu 20.04 machine running on AMD Ryzen Threadripper 3990X and Nvidia GeForce RTX 3090. Our CasTGAN source code is publicly available.[1]

A. DATASETS

Our CasTGAN is designed for synthesis of tabular data that can be typically found in the financial and healthcare sectors. We therefore use tabular mixed datasets characterised by a combination of categorical and numerical features. We use four datasets where the task is the binary classification of the target label - Adult [47], Bank Marketing [48], Taiwan Credit [49] and Diabetes [50]. Meanwhile, we use the House Prices [51] and Cars [52] datasets for regression. We additionally highlight that binary columns in the datasets are handled as categorical variables. An overview of the datasets used is presented in Table 2. For synthesising data with CasTGAN and other baselines, we use 50% of the datasets' total number of samples for training the models. The remaining 50% of the data samples is dedicated for evaluating the generated synthetic output.

TABLE 2. Datasets used in this study.

Dataset	Samples	Num. features	Cat. features	Unique categories	Task
Adult	32561	6	9	104	Binary classification
Bank	45211	10	11	55	Binary classification
Credit	30000	14	10	79	Binary classification
Diabetes	253680	7	15	30	Binary classification
Cars	472336	5	8	2409	Regression
Housing	21613	17	2	72	Regression

B. BASELINES

We compare the synthetic output of CasTGAN against five state-of-the-art tabular generative adversarial network models: table-GAN [33], medGAN [34], CTGAN [38], cWGAN [35] and CTAB-GAN [36]. Given that some of the datasets that we use in this study were not evaluated previously by the existing methods, we selected the optimal hyperparameters recommended by the baseline methods' authors in this study.

C. HYPERPARAMETER SELECTION

We emphasise that while our framework comprises an ample number of hyperparameters, we state that our reasoning for refraining from fine-tuning our model's hyperparameters in our experimental results is twofold. First, an extensive hyperparameter tuning would need to be conducted on a data-level basis and on a criteria level. To demonstrate that our framework is compatible with any dataset, we conduct our experimental analysis and train CasTGAN on all the datasets in this work using the same set of parameters in Appendix A-A. The choice of these parameters is based on preliminary, yet limited experimentation of our framework which yielded satisfactory and promising synthetic output during the conceptualisation phase. From a criteria-level perspective, we highlight that the model settings that attain the best performance on an evaluation criterion do not necessarily improve the synthetic output on all performance evaluation fronts. For instance, there might exist a hyperparameter trade-off between maximising the machine learning utility and improving the univariate feature representation. Second, given that we refrain from conducting hyperparameter tuning for the benchmark models and instead adopt the recommended hyperparameters of the benchmarks across all datasets, the comparison between our method and the baselines can be regarded as fair and equitable, provided that no extensive tuning of CasTGAN is undertaken correspondingly. Notwithstanding, we refer the reader interested in strategies for selecting hyperparameters for training CasTGAN on their own data to Appendix A-B.

D. EVALUATION CRITERIA

1) TRAIN ON SYNTHETIC, TEST ON REAL (TSTR)

We measure reliability of the synthetic output produced by CasTGAN by training machine learning models on the generated data. We fitted three machine learning models on the generated output - namely AdaBoost, random forest and logistic regression for classification tasks and AdaBoost, decision trees and Linear SVM for regression tasks. We then used the trained models to predict the target label of the test data and we report the precision-recall area under curve (PR-AUC) for binary classification tasks and the R^2 score for regression tasks. Since our main objective is to measure the machine learning utility of the generated data rather than assessing the individual performance of each classifier, we average the metrics produced by the three machine learning models.

2) UNIVARIATE DISTRIBUTIONS

We also assess the extent at which the individual features generated by CasTGAN resemble the features of the real data. As we quantitatively analyse how well our model learns the univariate feature distributions, we first one-hot encode and normalise the synthetic and real datasets. We calculate the dimension-wise mean of the individual features of the synthetic output and training data and report the RMSE between the real and the synthetic output. Additionally, we report the Kolmogorov-Smirnov two-sample test score [53] between the real univariate features and the synthetic ones.

3) CORRELATION AND DIVERSITY

Measuring the validity of the synthetic output represents a significant challenge for GAN frameworks. In computer vision applications of GANs, the synthetic images can be in some cases distinguished from the real images by a human observation of irregularities in the output such as pupil

[1] https://github.com/abedshantti/CasTGAN

orientation in human eyes and out-of-place pixels. In tabular GANs, similar challenges exist as there is no standard approach in the literature for quantifying the proportion of invalid samples in the synthetic output. For instance, a common example in the tabular synthesis literature is highlighting how an entry such as gender = "Female" and diagnosis = "Prostate Cancer" is by definition an invalid record, as no such entry exists in the original data, nor can a female be diagnosed with prostate cancer by a physician. In financial datasets, such nested relationships also exist between features if for example we inspect a dataset with a city column and a country column. In such cases, a record with city = "Buenos Aires" and country = "Malta" is by definition invalid. Meanwhile, entries with city = "Alexandria" and country = "United States" is valid as Alexandria exists in the United States even though it is more commonly attributed to the country "Egypt". It is for this reason that quantifying the invalid output generated by tabular GANs is no easy task, even in the presence of domain knowledge.

While a GAN model needs to ensure that its synthetic output is as valid as possible, there also needs to be some considerations for the diversity of the generated output. As such, the generative models should not significantly restrict the possible feature combinations between the different categorical features. Ensuring the diversity of the synthetic output enables the model to be less deterministic and increases its robustness against privacy attacks that aim to identify sensitive information. Therefore, the GAN model should be encouraged to explore unique feature combinations as long as such combinations can be considered valid.

Given that public datasets are used in this work for the purpose of reproducability, we do not have the full domain knowledge for these datasets, thus, we propose an alternative method for quantifying the validity of the synthetic data. First, we consider calculating the difference in feature correlations between the training data and the fake data. For computing the correlations between numerical features we use the Pearson's correlation coefficient, while the Cramer's V measure is used for capturing the correlation between categorical features. The correlation score is found by calculating the root mean squared error (RMSE) score between the elements of the triangular matrix of the synthetic dataset and the real dataset.

For measuring the diversity of the categorical output, we propose a new metric - Unique Pairwise Categorical Combinations (UPCC). In essence, we count the total number of unique interactions between any pair of categorical features in the dataset. For instance, in the Adults dataset the combinations [education = "Bachelors" and marital-status = "Never-married"], [education = "Bachelors" and sex = "Male"] and [marital-status = "Never-married" and sex = "Male"] each counts as a single pairwise combination, regardless of how many times they appear in the data. A reliable model therefore ensures that the UPCC of its output should be comparable to that of the original data. Subsequently, the UPCC Ratio is the number of unique pairwise combinations of synthetic output divided by the total number of unique combinations of the training data.

Finally, we estimate the validity of the model's output by dividing the correlation RMSE score of the model by the UPCC Ratio. We name this measure as the CORDV score. A lower CORDV score indicates that the model is able to minimise the difference in feature correlation between its synthetic output and the real data, while simultaneously not impeding its ability in generating unique feature combinations. Meanwhile, a worse generative model is reflected by a greater CORDV score, indicating that the model poorly captures the correlations while potentially restricting the uniqueness of the categorical pairs.

4) WHITE-BOX PRIVACY ATTACKS

Traditionally, white-box membership inference attacks on GANs assume that the attacker has access to the synthetic data and at least one generative component of the model. In this work, we formulate white-box privacy attacks in a different setting. We highlight that while using multiple auxiliary learners help in generating more realistic and reliable synthetic output, the use of multiple auxiliary learners leads to a more susceptible model for privacy breaches by attackers.

In this work, we devise white-box attacks by assuming that an attacker has access to the trained auxiliary learners and attempts to reconstruct training samples through an iterative process of estimating a hidden feature. In essence, the attacker with the synthetic data will at a given time remove one column from the data, use the corresponding the auxiliary learner to predict the masked feature using the remaining features, and then replacing the masked column with the predicted output from auxiliary learners. In this setting, a single iteration refers to a walk-through over all the auxiliary learners and subsequently replacing all the columns in the dataset once.

For evaluating how effective such white-box attacks on our model, we control the training the of the auxiliary learners using a perturbation parameter ϵ. The perturbation parameter translates to the proportion of label samples that are modified when training the auxiliary learners prior to the GAN training. For an auxiliary learner corresponding to a numerical column X_i, we perturb the numerical variables such that perturbed variable for a given sample $\tilde{x}$ can be calculated as $\tilde{x}_i = x_i + \alpha x_i$, where α is a floating number randomly sampled from $[-1.0, 1.0]$. Meanwhile, we perturb the categorical features by randomly selecting a category from the list of all the unique categories of the said feature. In our analysis, we experiment with $\epsilon = 0.0$, implying that no perturbation takes place, and gradually increment this value to $\epsilon = 0.3$, implying that 30% randomly chosen samples for each auxiliary learner were perturbed prior to the auxiliary training.

Furthermore, we analyse whether an attacker possessing the original data preprocessing transformers has an additional advantage in recovering the training samples. We hypothesise that an attacker with access to the data transformations used for the auxiliary training can simply convert the synthetic to a data structure that aligns with the existing transformer. On the other hand, an attacker without access to the preprocessors needs to fit and transform the data independently before launching membership attacks. This is especially prevalent for categorical features, where transforming the categories into ordinal encodes that do not match the ones learned by the auxiliary learners might lead to less effective attacks.

VI. RESULTS

A. MACHINE LEARNING UTILITY

We use the synthetic data generated by CasTGAN for fitting machine learning classification and regression supervised models on the six datasets and compare our performance on the test set against models fitted on the training set and models fitted on the five synthetic output of the five baseline methods. Additionally, we compare the performance of our model and the other baselines against the real training datasets used for fitting the predictive models, which we refer to as *Identity*. We note that we were unable to run the highly-dimensional Cars dataset on CTAB-GAN to exceedingly large memory requirements attributed to representing very high-dimensional data in the convolutional neural network GAN-based approach. The results are computed in Table 3.

TABLE 3. Binary classification (PR-AUC score) and regression (R^2 score) evaluation on the test sets.

	Adult	Bank	Credit	Diabetes	Cars	Housing
Identity	0.7744	0.6085	0.5374	0.3980	0.7604	-0.3696
table-GAN	0.2225	0.0954	0.2084	0.1585	-2.0148	-83.1326
medGAN	0.5777	0.2729	0.3050	0.3241	**0.7491**	**-0.3845**
CTGAN	0.6932	0.4805	0.4558	0.3711	0.5327	-0.6533
cWGAN	0.3030	0.2692	0.2166	0.3325	-319.0963	-1.1054
CTAB-GAN	**0.7192**	0.4920	0.4911	0.3589	-	-0.7284
CasTGAN	0.6718	**0.5657**	**0.4995**	**0.3866**	0.5566	-0.4433

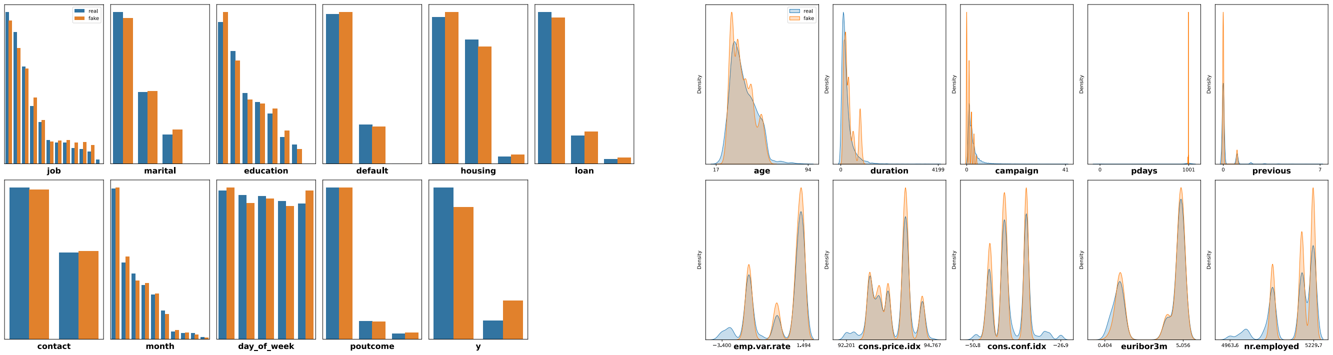
From Table 3 we can observe that the TSTR metrics for our CasTGAN is consistently within the best performing synthetic output, outperforming all the baselines on three out of six datasets. On the Bank dataset, we can observe that CasTGAN's PR-AUC score of 0.5657 ranks closer to the classification models trained on the Bank real training data with a score of 0.6085, than the second best classification by CTAB-GAN with a score 0.4920. Similarly, CasTGAN falls 0.0114 short of the PR-AUC score exhibited by the real data on the Diabetes dataset, whilst outperforming the second best synthetic model, CTGAN, with a PR-AUC difference of 0.0155. We also observe that CasTGAN also ranks first among the GAN models on the Credit dataset, despite being closely challenged by CTAB-GAN. For the datasets which CasTGAN did not achieve the highest machine learning utility, the results in Table 3 show that CasTGAN narrowly underperformed against two of the baselines on the Adult dataset. Furthermore, we note that medGAN demonstrated the best results on regression datasets, whereas, synthetic data from CasTGAN followed as the second best in terms of the R^2 score on the regression datasets. In general, the prediction results on the test sets suggest that synthetic data produced by CasTGAN is well suited to fitting machine learning models, as the predictive performance is comparable to training on the real data, and in-line with the best performing state-of-the-art tabular GANs.

B. UNIVARIATE SIMILARITY

It is imperative that synthetic data generation techniques need to emulate the distribution of the features of the real data. One method for qualitatively evaluating the statistical similarity between the real data and the synthetic data is to visually compare the distributions for categorical and numerical attributes. Subsequently, we choose to display the comparison between the synthetic and real features for the Bank dataset, as it consists of a diverse and heterogeneous set of features. The depiction is demonstrated in Figure 4.

From Figure 4, we can observe how well CasTGAN performs in approximating the distributions for the categorical and numerical attributes. For categorical variables in Figure 4a, it is evident that CasTGAN preserves the frequency of unique categories under each discrete variable. It can also be observed that our framework can successfully represent and sample the less frequent categories in the dataset. For the numerical features in Figure 4b, we can find that the synthetic data distribution closely approximates the numerical distribution the real data. While adopting the Variational Gaussian Mixture models in CasTGAN helps in improving the density estimation of numerical features, as indicated by the number of peaks and the extensive coverage of the numerical bounds, we notice that approximation is slightly conservative, thus, does not represent extreme values in the real data's numerical range to avoid violating boundary constraints and generating invalid records. The combined results of the numerical and categorical features distributions of the synthetic data from CasTGAN, in addition to the reasonable sampling of less frequent features, as demonstrated in Figure 4, signifies that our framework clearly does not suffer from mode collapse.

To quantitatively analyse how well CasTGAN represents the feature distributions of the original data, we compare the Euclidean distance RMSE and Kolmogorov-Smirnov statistic between the synthetic data and the real data in Table 4. We find that CasTGAN performs considerably the best in terms of the Euclidean distance RMSE on three of the six benchmark datasets: Adult, Bank, and Cars, while ranking comparatively to the best performing GANs on the three remaining datasets. It can also be observed that while CasTGAN only achieves the best KS statistic on the Cars dataset, and often trailing only slightly behind CTAB-GAN on the other datasets, the KS statistic by our model regularly maintains a small value, which can be interpreted as the higher likelihood of CasTGAN's synthetic data to come from the same distribution as the real data. Generally, we note that our CasTGAN, along with medGAN and CTAB-GAN dominate the dimension-wise statistical similarity test. It is also apparent that CasTGAN represents the features of the Adult and Cars datasets particularly well, while performing comparatively on the remaining datasets. We can therefore



(a) Univariate categorical feature-wise distribution comparison between the real data (in blue) and the CasTGAN synthetic data (in orange). The counts of the categories are log-scaled for intuitively representing infrequent categories.

(b) Univariate numerical feature-wise distribution comparison between the real data (in blue) and the CasTGAN synthetic data (in orange). The distributions are plotted by applying Gaussian kernel density estimation.

FIGURE 4. Discrete and continuous univariate features distribution plots for the bank dataset.

TABLE 4. Univariate dimension-wise statistical comparison showing the Euclidean root mean squared error (Euc. RMSE) and the Kolmogorov-Smirnov two-sample test score (KS statistic).

(a) Adult, Bank and Credit datasets

	Adult		Bank		Credit	
	Euc. RMSE	KS statistic	Euc. RMSE	KS statistic	Euc. RMSE	KS statistic
table-GAN	0.2249	0.6499	0.3485	0.5969	0.3218	0.8794
medGAN	0.0599	0.1290	0.0636	0.1963	0.0570	0.3038
CTGAN	0.0377	0.1502	0.0415	0.1076	0.0786	0.1715
cWGAN	0.0406	0.1111	0.0616	0.5536	**0.0167**	0.1906
CTAB-GAN	0.0186	**0.0812**	0.0396	**0.0871**	0.0335	**0.0959**
CasTGAN	**0.0111**	0.0908	**0.0276**	0.1470	0.0283	0.1327

(b) Diabetes, Cars and Housing datasets

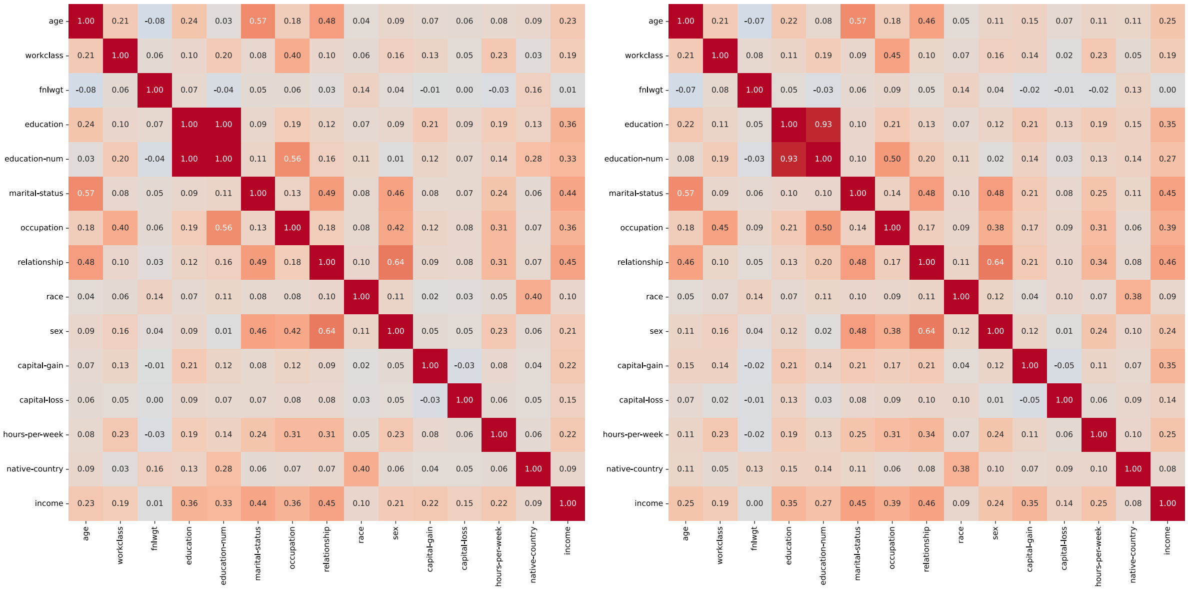
	Diabetes		Cars		Housing	
	Euc. RMSE	KS statistic	Euc. RMSE	KS statistic	Euc. RMSE	KS statistic
table-GAN	0.4087	0.4247	0.0531	0.7088	0.1476	0.6606
medGAN	**0.0188**	**0.0447**	0.0124	0.1323	**0.0141**	0.1812
CTGAN	0.0767	0.1024	0.0067	0.0785	0.0571	0.1682
cWGAN	0.1133	0.2809	0.1065	0.2890	0.1096	0.2956
CTAB-GAN	0.0648	0.0798	-	-	0.0150	**0.0989**
CasTGAN	0.0834	0.0780	**0.0019**	**0.0457**	0.0184	0.1651

deduce that CasTGAN is particularly useful for datasets with a greater number of unique categories.

C. OUTPUT VALIDITY

In addition to ensuring univariate similarity, it is equally fundamental to evaluate how well the synthetic models preserve the interactions between the different features of a dataset. Inspecting the correlations between the data features of the real data and comparing the correlations with the synthetic data representation can indicate whether the generative data models are capable of simulating the relationship between the data variables from the real data representation. Figure 5 depicts the correlation matrices for the Adult real training data and the synthetic Adult data generated by CasTGAN. We can see from Figure 5 that our model notably learns the correlations between the real data features during the training process. By comparing Figure 5a and Figure 5b, it is evident that there is only a marginal difference between the correlations of the real data and the CasTGAN synthetic data, the highest of which we note is a correlation difference of 0.14 between the ‘‘native-country’’ and ‘‘education-num’’ features. Otherwise, the model successfully emulates both the magnitude of the correlations and the sign, as to whether the feature correlations are positive or negative.

As emphasised in the motivation for this work, there is a need to establish evaluation frameworks that can critically assess the *realistic-ness* of the synthetically generated data. We therefore quantify the validity of the output by considering the number of unique pairwise categorical combinations (UPCC), the correlation error between the synthetic and the real data, and the CORDV score, which is essentially the correlation divided by the UPCC ratio. From Table 5,



(a) Real data correlation matrix

(b) CasTGAN synthetic data correlation matrix

FIGURE 5. **Adult dataset correlation map plots for the real and the synthetic data generated by CasTGAN. Larger absolute values indicate a stronger correlation, either positively or negatively.**

TABLE 5. **Diversity and correlation comparison demonstrating the number of unique pairwise categorical combinations (UPCC), the correlation root mean squared error (Corr. RMSE) and the CORDV score.**

(a) Adult, Bank and Credit datasets

	Adult			Bank			Credit		
	UPCC	Corr. RMSE	CORDV	UPCC	Corr. RMSE	CORDV	UPCC	Corr. RMSE	CORDV
Identity	3004			1244			1426		
table-GAN	1168	0.2010	0.5170	461	0.2516	0.6789	878	0.3001	0.4874
medGAN	3769	0.1708	0.1361	1233	0.1371	0.1383	1971	0.1826	0.1321
CTGAN	**3435**	0.0570	0.0498	**1306**	**0.0749**	**0.0713**	**2501**	0.1003	**0.0572**
cWGAN	1205	0.1899	0.4734	993	0.2043	0.2559	734	0.2612	0.5075
CTAB-GAN	2404	0.0718	0.0897	1255	0.1128	0.1118	1963	**0.0804**	0.0584
CasTGAN	2369	**0.0368**	**0.0466**	1163	0.0822	0.0879	1191	0.1130	0.1353

(b) Diabetes, Cars and Housing datasets

	Diabetes			Cars			Housing		
	UPCC	Corr. RMSE	CORDV	UPCC	Corr. RMSE	CORDV	UPCC	Corr. RMSE	CORDV
Identity	420			62872			85		
table-GAN	134	0.1482	0.4646	28	0.4015	901.5202	62	0.3493	0.7890
medGAN	**420**	0.0891	0.0891	165522	0.2734	0.1039	70	0.2220	0.2696
CTGAN	**420**	0.0575	0.0575	**128405**	0.1677	**0.0821**	**140**	0.1755	0.1065
cWGAN	392	0.1135	0.1216	739	0.3409	29.0038	69	0.2322	0.2861
CTAB-GAN	**420**	0.0512	0.0512	-	-	-	107	0.1384	0.1100
CasTGAN	406	**0.0490**	**0.0507**	31782	**0.1351**	0.2672	83	**0.0880**	**0.0909**

we can observe how the different generative methods rank among the three aforementioned metrics. First, we note that CTGAN and medGAN perform well in generating a large number of unique feature combination, in most cases, even more than the number of combinations that can be found in the training data. This is particularly impressive for the Cars dataset, where both models managed to generate more than twice the number of categorical combinations of the training set. Similarly, we also observe that CTAB-GAN performs relatively well in exploring diverse categorical combinations. Meanwhile, we notice that our CasTGAN is more conservative when it comes to generate unique categorical combinations. For all the datasets, CasTGAN produces a marginally lower number of pairwise combinations than can be typically found in the training set. Moreover, the UPCC can be a good indicator of mode collapse and this is reflected by the significantly low UPCC values for table-gan and cWGAN, where it can be deduced that these models generated a limited number of modes for some categories.

In contrast, it can be observed from Table 5 that CasTGAN generally outperforms the other baselines in capturing the feature correlations of the datasets. The lower correlation RMSE score entails that CasTGAN prioritises the representation of correlations and feature interdependence in the real data. Meanwhile, the CORDV score aims to quantify the trade-off between the diversity and the proximity of the synthetic data to the real data. We observe from the results that the highest CORDV scores are evenly split among our CasTGAN and CTGAN.

Despite lacking the full domain knowledge in our datasets, we nonetheless measure the synthetic invalidity on the Adult, Cars and Housing datasets as follows:

- *Adult dataset*: we use the fields "relationship" and "sex" to calculate the number of invalid records. We posit that if "relationship" = "Husband", then the sex feature needs to be set to "Male". Likewise, the "relationship" = "Wife" needs to align the gender field assigned as "Female". This is not based on our assumptions, but rather running exploratory data analysis on the training set confirms that the records are matched in such a manner.
- *Cars dataset*: we use the fields "make" and "model" and classify a synthetic sample as invalid if the synthetic car's "model" does not in fact belong to the "make" that can be found in training data.
- *Housing dataset*: the housing dataset consists of the fields "year built" and "year renovation". Logically, a property cannot be renovated before it was built, and further inspecting the data indeed confirms that there are no observations with "year renovation" that precedes "year built".

Based on the aforementioned fronts, the ratio of invalid records generated by CasTGAN and three baseline methods are demonstrated in Table 6. We chose not to include tableGAN and cWGAN in the comparison as their synthetic output was found to be characterised by major mode collapse.

TABLE 6. An outline of the ratio of invalid synthetic records of the baseline GANs and our model.

	Adult	Cars	Housing
medGAN	29.05%	91.11%	59.29%
CTGAN	9.82%	42.89%	54.75%
CTAB-GAN	4.14%	-	44.42%
CasTGAN	**0.67%**	**27.82%**	**22.98%**

From Table 6 we can observe that CasTGAN remarkably reduced the number of invalid synthetic records of the Adult dataset. Our method also significantly decreased the number of invalid records in the Cars dataset. This resembles a major improvement from CTGAN, while noting the challenging nature of modelling the Cars dataset due to the large number of categories present. Furthermore, it is evident that CasTGAN outperforms the other generative approaches on the numerical features of the Housing dataset.

D. ROBUSTNESS AGAINST PRIVACY ATTACKS

For conducting white-box privacy attacks, we set the number of attacking iterations to five, where each feature in the synthetic data is updated five times based on the output of the auxiliary learners. Moreover, the membership attacks are launched on 10% of the total number of overall samples. We highlight that the ratio of attacked samples does not impact the evaluation of the robustness of our approach as we only compute the attack distance metrics with respect to the attacked samples. The Euclidean distance of the attacked samples to the training data and to the synthetic data prior to the membership attacking is computed in Table 7.

From Table 7, it can be observed that the perturbation coefficient ϵ greatly impacts the closeness of attacked samples to the training data. For unperturbed and minimally perturbed data features, it can be noticed that the attacked samples are relatively close to the training samples, indicating that the attackers might succeed in recovering training datapoints. We observe that the proximity to the training samples increases for greater ϵ values, which demonstrates the additional privacy guarantees that can be provided when altering the labels. We additionally notice how access to the data processors gives a major advantage to the attackers attempting to recover the training samples. This holds true especially for the Adult dataset, where using the trained label encodings of the auxiliary learners lead to more targeted attacks that are even closer to the training samples than attackers on unperturbed data with no access to the data preprocessors. Another interesting observation is that the attacks on the Housing dataset are greatly impacted by incremented in ϵ, which is plausible, given that the dataset mainly consists of numerical features.

In addition to the proximity to the training samples, we also investigate whether perturbing the labels of the auxiliary learners can contribute to a reduction in the quality of unattacked synthetic data as demonstrated in Table 8. For the Adult, Bank and Credit datasets it can be evident that applying perturbations insignificantly impacts the the evaluation metrics of the synthetic datasets. We observe that the PR-AUC scores on the test data and KS statistic for univariate distributions are minimally influenced by the changes in ϵ. In contrast, it appears that perturbing the data impacts the CORDV scores as a result of the correlation errors between the synthetic and the real datasets. Interestingly, applying perturbations on the Housing dataset appear to improve the quality of the synthetic output in addition to increasing its robustness against white-box privacy attacks.

E. IMPACT OF AUXILIARY LEARNERS LOSS

We further analyse the impact of the auxiliary learners on the quality of the synthetic data samples produced by CasTGAN. To this end, we conduct experiments on our model by tuning the auxiliary loss coefficient parameters $\lambda_{AL_1}, \lambda_{AL_2}, \ldots, \lambda_{AL_M}$. This is implemented by adjusting λ_{AL_1} and λ_{AL_M}, as the auxiliary loss coefficients in between are linearly and equidistantly scaled between the loss coefficients of the first and final auxiliary learners. Given that there is an indefinite number of auxiliary loss coefficients combinations and a diverse set of evaluation metrics and datasets, we conduct our the analysis on the Adult dataset by experimenting with a set of auxiliary loss coefficient values: $\{0, 0.25, 0.5, 0.75, 1\}$. As the focus of our work is improving the *realistic-ness* and maximising the number of semantically valid synthetic records, we quantify the

TABLE 7. White box privacy attacks on the auxiliary learners proximity measures.

		Access to only AL		Access to AL and Preprocessing	
		Euc. to train	Euc. to pre-attack syn	Euc. to train	Euc. to pre-attack syn
Adult	$\epsilon = 0.0$	1.2202	1.4437	0.1256	0.3652
	$\epsilon = 0.1$	1.4250	1.7289	0.2025	0.7348
	$\epsilon = 0.2$	1.4747	1.8288	0.6848	1.4039
	$\epsilon = 0.3$	**2.4468**	2.5217	**0.7413**	1.2509
Bank	$\epsilon = 0.0$	2.2275	2.4975	1.2225	1.7366
	$\epsilon = 0.1$	3.0541	3.0272	2.4020	2.4182
	$\epsilon = 0.2$	5.6174	5.9515	5.3565	5.8131
	$\epsilon = 0.3$	**6.7180**	7.1127	**6.7095**	7.0654
Credit	$\epsilon = 0.0$	0.3740	0.6550	0.1157	0.3179
	$\epsilon = 0.1$	0.6196	0.7916	0.1094	0.2804
	$\epsilon = 0.2$	0.3918	0.5797	0.0732	0.2773
	$\epsilon = 0.3$	**1.5533**	1.7834	**0.6349**	0.8470
Housing	$\epsilon = 0.0$	0.3696	0.4905	0.2026	0.4132
	$\epsilon = 0.1$	4.5976	4.7431	4.4731	4.5013
	$\epsilon = 0.2$	16.2106	16.6204	16.1899	16.5996
	$\epsilon = 0.3$	**28.6935**	28.8148	**28.6990**	28.8205

TABLE 8. Impact of auxiliary learners imputation on the synthetic output.

		TSTR	KS	CORDV
Adult	$\epsilon = 0.0$	0.6718	**0.0908**	0.0467
	$\epsilon = 0.1$	0.6683	0.1173	**0.0380**
	$\epsilon = 0.2$	**0.6731**	0.1560	0.0393
	$\epsilon = 0.3$	0.6647	0.1346	0.0425
Bank	$\epsilon = 0.0$	**0.5657**	0.1470	0.0879
	$\epsilon = 0.1$	0.5384	**0.1020**	**0.0847**
	$\epsilon = 0.2$	0.5551	0.1253	0.0868
	$\epsilon = 0.3$	0.5475	0.1130	0.0960
Credit	$\epsilon = 0.0$	0.4995	0.1327	**0.1353**
	$\epsilon = 0.1$	**0.5140**	**0.1069**	0.1504
	$\epsilon = 0.2$	0.4722	0.1170	0.1434
	$\epsilon = 0.3$	0.5095	0.1103	0.1822
Housing	$\epsilon = 0.0$	-0.4430	0.1651	0.0909
	$\epsilon = 0.1$	-0.5602	0.2185	0.0864
	$\epsilon = 0.2$	-0.3257	0.1554	0.0961
	$\epsilon = 0.3$	**-0.2966**	**0.1381**	**0.0828**

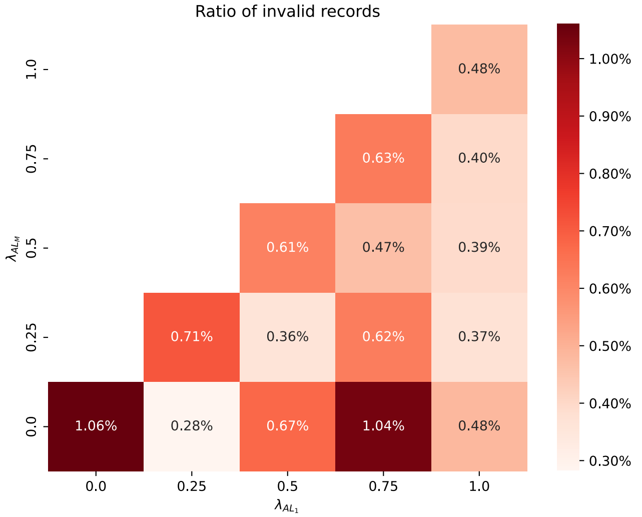


FIGURE 6. Proportion of invalid synthetic records of the Adult dataset generated by CasTGAN using various auxiliary loss coefficient settings.

number of invalid records of the synthetic samples as per the description of invalid records of the Adult dataset defined in Section VI-C. Figure 6 demonstrates the fraction of invalid records generated by CasTGAN for various auxiliary loss coefficients.

As shown in Figure 6, adjusting the auxiliary loss coefficients is reflected by a modest change in the ratio of invalid synthetic records of the Adult dataset. As demonstrated, it appears that using a fixed auxiliary loss coefficient across the auxiliary learners by setting the value λ_{AL_1} and λ_{AL_M} generally increases the number of invalid records. Meanwhile, it can be observed that having a smaller difference between λ_{AL_1} and λ_{AL_M} leads to an improved performance in contrast to when the difference is 0.5 or greater in general. We can notice that the lowest fraction of invalid synthetic observations is obtained by setting $\lambda_{AL_1} = 0.25$ and $\lambda_{AL_M} = 0$. By comparing the ratio of invalid records in Table 6 and Figure 6, it is clear that the default auxiliary loss coefficient parameters that we used in our experimental setup do not yield the most optimal results, whereas further hyperparameter tuning can in practice help reduce the number of invalid records. Meanwhile, it is evident that not relying on auxiliary learners by setting the loss coefficients to zero contributes to the highest fraction of invalid synthetic records. However, we can observe that the ratio of invalid records generated when using loss coefficients of 0, thereby nullifying the effect of auxiliary learners, is nevertheless lower than the proportion of the invalid synthetic records generated by the baseline models, as displayed in Table 6. This suggests that while the improved quality of synthetic records can be to a great extent be attributed to the design of the cascaded generator architecture and employment of the WGAN-GP, the use of auxiliary learners and fine-tuning the auxiliary loss coefficients can further contribute to the generation more semantically valid records.

VII. DISCUSSION

Data in tabular form is widely used across organisations in various domains for decision support systems. In such systems, data mining techniques and statistical models are employed to enhance the predictability of significant events and to gain a deeper insight into user behaviour. Moreover, mixed-type tabular data is commonly used for facilitating knowledge exchange between domain experts and stakeholders. Given the privacy concerns associated with sharing confidential information, in addition to the scarcity of available data to share across organisations, there has been a growing interest in tabular data synthesis using generative adversarial networks, in which realistic synthetic data can be safely shared across operational units. In literature, the focus of generative models has been primarily directed towards generating synthetic data that yields acceptable machine learning utility and exhibits statistical properties which are similar to the real data. While these objectives are relatively significant, an underlying limitation is that the semantic integrity of the synthetic output is not thoroughly examined. Given that GANs attempt to approximate the data distributions loss minimising functions for the generator and the discriminator, it is imperative that GAN-based synthesis models are unable explicitly distinguish between semantically valid and invalid data records.

To this end, the cascaded tabular GAN architecture we propose in this study contributes to the reduction of the number of invalid generated observations by dedicating a generator for every feature of the dataset. The sequential cascading of generator passes incomplete synthesisation of the data to subsequent generators, in which each generator attempts to predict its target feature from the incomplete feature space it receives from the preceding generator. As the generator fills the remaining feature space with noise, the output is queried against the discriminator and the discriminator loss for the queried sample is backpropagated to the generator for improving its overall generation. For obtaining meaningful feedback to the generator's designated feature, an auxiliary learner is coupled to the generator that calculates the prediction loss for the specific feature and propagates the loss to the generator, such that it can learn to reduce the auxiliary learner's loss during GAN training. In such an architecture, the GAN attempts to learn the dependencies between the dataset's features, and thereby gradually reduces the losses for the components of the GAN during the training process. Hence, the aim of CasTGAN is to increase the semantic integrity of the synthetic and reduce the invalid records that are generated. This is highly significant as the generation of invalid records may have adverse effects on knowledge sharing, potentially fostering an inaccurate comprehension of the data among stakeholders and other external entities granted access to the synthesized data.

Notably, our results indicate where our proposed generative model has remarkably performed. First, the machine learning utility results demonstrate that the synthetic output from CasTGAN is particularly useful for constructing predictive models, as reflected by the classification and regression performance metrics. The results show that CasTGAN is competitive with the state-of-the-art tabular GANs, outperforming them on some datasets, and barely falls short to the predictive performance obtained by training on the real data. Regarding the univariate variable properties, we visually demonstrate the synthetic output has a striking similarity to the the real data numerical and categorical variables. Furthermore, it can be observed that quantitative similarity analysis indicates a very close similarity between the data distributions of the real and the CasTGAN synthetic output. In accordance to addressing the gap in literature for better exploring the semantic integrity of the synthetic data, we evaluate the output validity of our synthetic data on several fronts. The correlation mapping shows that the synthetic data from our model exhibits a strong resemblance to the real data used for training. We also notice that CORDV metric which we design for measuring the correlation error as a fraction of the unique pairwise combination successfully shows that our approach is well capable of capturing the dependencies between the data features. In general, the performance of CasTGAN for machine learning usability, statistical similarity and correlation and diversity was mostly comparable with CTGAN and CTAB-GAN. This can be attributed to the use of WGAN-GP in the case of CTGAN, or classification loss and information loss in the case of CTAB-GAN, both of which contribute to the training stability and the mitigation against mode collapse. Meanwhile, calculating the ratio of invalid records on a number of datasets demonstrates that CasTGAN excels in reducing the number of invalid data observations in comparison to the existing tabular GAN approaches, thereby improving the potential of using the synthetic data for knowledge exchange. Further analysis unveils that applying perturbations on the auxiliary learners can increase the robustness of our model against privacy attacks without notably sacrificing our model's synthetic output quality.

We also note that using the auxiliary learners leads to a more conservative approach for the GAN training process. The reduction of the invalid records comes at the expense of reducing the number of unique pairwise combinations of the data categories that CasTGAN can synthesise, and additionally not fully simulating numerical values around the boundaries for numerical features. Nevertheless, we highlight that the reduction of invalid records and improving the model's representation of feature dependency was the focus of this work, and that CasTGAN demonstrates success in these two aspects.

VIII. CONCLUSION

In this work, we presented CasTGAN as a generative framework for creating synthetic tabular data samples that are representative of the real data attributes. Our motivation for this work stems from the need for realistic tabular data that can be exchanged amongst experts, while focusing on the reliability and the sensitivity of such information. We therefore directed our focus towards generating fake output that capture the correlations and interdependence between the data features. We demonstrated that our cascaded generator architecture supported by auxiliary learners are able to generate realistic output given highly dimensional and largely imbalanced tabular datasets. Our results indicate that CasTGAN is capable of significantly reducing the number of invalid records while exhibiting strong statistical and correlational similarities to the real data. We further evaluated the robustness of our model against targeted privacy attacks and showed that perturbing the auxiliary learners by a small scale can mitigate against attacks aiming to recover the real data samples.

Given the challenging nature of generating realistic synthetic tabular data, there are several paths for future work. This work can be extended by incorporating additional data types within the tabular data generation such as free text and timestamps. Due to cascaded architecture and the presence of multiple auxiliary learners, we point out that our framework does not offer improvements to the training speed over the existing tabular GAN models, which presents a potential opportunity for future optimisation efforts. Moreover, we intend to explore how our approach can generate more diversified combination of categories, while maintaining its ability in minimising the number of invalid data records.

APPENDIX A IMPLEMENTATION DETAILS

A. HYPERPARAMETERS

TABLE 9. CasTGAN hyperparameters used for all datasets.

Hyperparameter	Value
Epochs	300
Batch size	512
Generators noise input dimension	128
Generators noise input distribution	$\sim \mathcal{N}(0, 1)$
Generators primary networks hidden sizes	[128, 64]
Generators primary networks inter-hidden layer activations	LeakyReLU
Generators primary networks activation normalizer	Layer Normalization
Generators primary networks numerical feature activation layer	tanh
Generators primary networks categorical feature activation layer	gumbel softmax$(\tau = 0.8)$
Generators secondary networks hidden sizes	[32]
Generator optimizer	Adam$(\alpha = 2 \times 10^{-4}, \beta_1 = 0.50, \beta_2 = 0.99)$
Discriminator primary networks hidden sizes	[256, 128]
Discriminator inter-hidden layer activations	ReLU
Discriminator activation normalizer	Layer Normalization
Discriminator optimizer	Adam$(\alpha = 2 \times 10^{-4}, \beta_1 = 0.50, \beta_2 = 0.99)$
Discriminator real input noise perturbation	$\sim \mathcal{N}(0, 0.01)$
Discriminator updates per generators update	1
Auxiliary learners boosting iterations	150
Auxiliary learners number of leaves	31
Auxiliary learners learning rate	0.10
Auxiliary learners early stopping	10 rounds
Wasserstein loss gradient penalty λ_{GP}	10
Auxiliary learner 1 loss coefficient λ_{AL_1}	0.75
Auxiliary learner M loss coefficient λ_{AL_M}	0.10

B. STRATEGIES FOR HYPERPARAMETER CONFIGURATION

There are various hyperparameters in CasTGAN, and therefore selecting the optimal hyperparameters is a non-trivial task. Defining what is considered as optimal also poses a challenge, as there may be a trade-off between enhancing the suitability of synthetic output for machine learning model fitting and increasing the semantic validity of the synthetic data. In light of this, we outline the parameters that have some significant impact on training our GAN model, and we describe how they can be tuned for optimising the performance. The parameters with less significance in correspondence to the quality of the synthetic data can utilise the default values from Appendix A-A.

1) EPOCHS

Tuning the number of training iterations can offer improvements to the quality of the synthetic output. For smaller datasets, it can be sufficient to train for a smaller number of iterations, in which the training completes faster. Meanwhile, larger datasets may require a greater number of epochs to demonstrate an improved synthetic output. It is recommended to train for a minimum of 100 epochs, though training the GAN for too long can lead to the memorisation of records by the GAN, inducing potential data records leakage and susceptibility to privacy attacks.

2) BATCH SIZE

Batch size refers to the number of samples stored in memory for updating the internal model parameters at every training iteration. A larger batch size is more suitable for datasets with a greater number of features and categories, whereas datasets with fewer features can be sufficiently trained with a smaller batch size. It is worth noting that a larger batch size contributes to a relatively slower training of the GAN and can contribute to a poor model convergence if not trained for a sufficient number of epochs.

3) GENERATORS NOISE INPUT DIMENSION

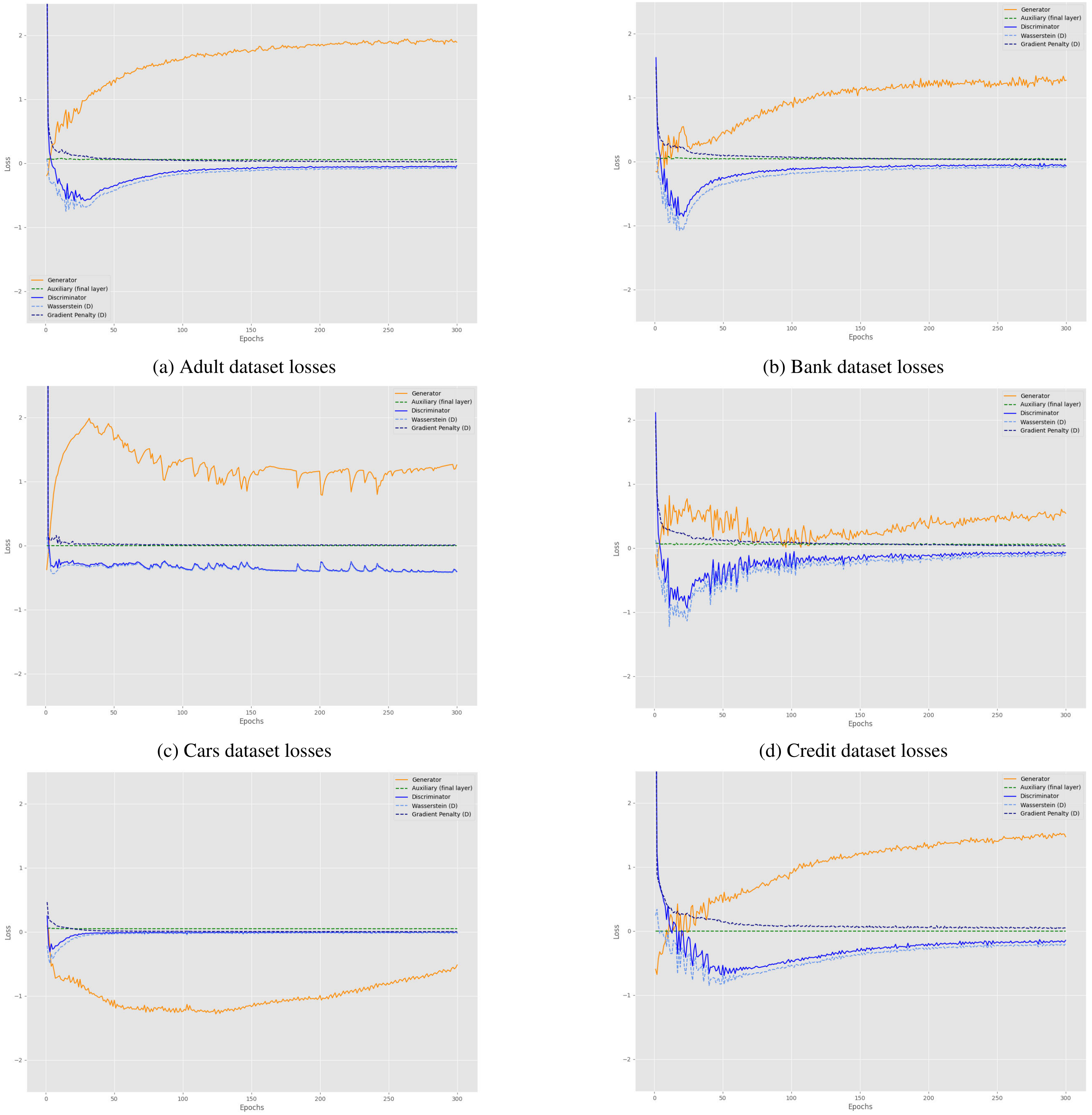
The optimal size of the noise vector used as the input for the generators is best determined through experimentation. Wasserstein GANs commonly exhibit a higher resolution of the synthetic output through the utilisation of larger noise vector [54], though the significance of input noise size is largely dependent on the data and the other model parameters.

4) GENERATORS PRIMARY NETWORKS HIDDEN SIZES

The number and the sizes of the hidden layers can significantly impact the quality of the synthetic output from the generators. Despite designating a generator for each data feature, the generators employ a common hidden layer configuration rather than a specific configuration for each generator. As each generator is responsible for generating one primary feature, the linear layers do not need to be overly sophisticated.

5) DISCRIMINATOR PRIMARY NETWORKS HIDDEN SIZES

The architecture of the discriminator should be able to foster the competitiveness between the discriminator and the generator, hence maintain the adversary throughout the GAN training. If the neural network architecture is relatively simpler than that of the generators, the discriminator will poorly distinguish between the synthetic and the real samples. Meanwhile, a complex discriminator architecture might suppress the learning of the generators, thus, degrade the synthetic output quality of the generators. Therefore, the number and the size of the discriminator's hidden layers should be comparable to those employed by the generators.



(a) Adult dataset losses

(b) Bank dataset losses

(c) Cars dataset losses

(d) Credit dataset losses

(e) Diabetes dataset losses

(f) Housing dataset losses

FIGURE 7. **CasTGAN training loss plots.**

6) DISCRIMINATOR UPDATES PER GENERATORS UPDATE

It might be worthwhile to adjust the number of discriminator iterations for each generator iteration, if it helps in converging the losses during training. This might be particularly the case if the discriminator is visibly weaker than the generator. If there is no major disparity between the discriminator's architecture and the generators' hidden layers, then there is no proven benefit in employing more than 1 discriminator gradients update for each step of the generators training [55].

7) WASSERSTEIN LOSS GRADIENT PENALTY (λ_{GP})

We find that using a Wasserstein loss gradient penalty of 10 works well, based on the training stability during our preliminary model conceptualisation. This is supported by the authors of the original WGAN-GP paper [13], who found the gradient penalty of 10 works well on various datasets and GAN architectures. This is also in accordance to the baseline tabular GANs with WGAN-GP we compare our approach against [28], [35], whom also employ $\lambda_{GP} = 10$.

8) AUXILIARY LEARNER 1 LOSS COEFFICIENT (λ_{AL_1})

The first auxiliary learner in the cascaded generator is assigned with the largest loss coefficient among to ensure that the univariate statistical representation of the first features of the data conforms to the real data distributions. The particular value for the first auxiliary learner's coefficient is chosen such that large prediction errors attributed to the deviation in distributions are sufficiently penalised, without making it too large such that training instability is introduced to the generators.

9) AUXILIARY LEARNER M LOSS COEFFICIENT (λ_{AL_M})

The final auxiliary learner can benefit from low loss coefficient values to minimise the errors propagated to the preceding generators. The purpose of the auxiliary loss coefficient of auxiliary learner M is to reduce the correlation error in the synthetic data towards the end of cascaded generator, as opposed to the first generators that prioritise the univariate similarity. In a similar manner to the first auxiliary learner, this is a hyperparameter that is largely dependent on the dataset used for training and sampling of CasTGAN. It is however advisable to set λ_{AL_M} with a very small value if a small value is selected for λ_{AL_1}.

C. TRAIN ON SYNTHETIC, TEST ON REAL

See Tables 10–15.

TABLE 10. Adult dataset classification.

	Accuracy	ROCAUC	F1-score	PR-AUC
Identity	0.8550	0.9075	0.6663	0.7744
table-GAN	0.5617	0.4102	0.1376	0.2225
medGAN	0.7920	0.8038	0.4888	0.5777
CTGAN	0.8304	0.8751	0.6235	0.6932
cWGAN	0.7586	0.5887	0.0002	0.3030
CTAB-GAN	0.8400	0.8849	0.6542	0.7192
CasTGAN	0.8272	0.8779	0.6380	0.6718

TABLE 11. Bank dataset classification.

	Accuracy	ROCAUC	F1-score	PR-AUC
Identity	0.9084	0.9342	0.4978	0.6085
table-GAN	0.2149	0.3910	0.1670	0.0954
medGAN	0.8803	0.6492	0.1547	0.2729
CTGAN	0.8979	0.8474	0.3492	0.4805
cWGAN	0.8337	0.6864	0.1297	0.2692
CTAB-GAN	0.8970	0.8768	0.3550	0.4920
CasTGAN	0.8991	0.9187	0.5823	0.5657

TABLE 12. Credit dataset classification.

	Accuracy	ROCAUC	F1-score	PR-AUC
Identity	0.8201	0.7685	0.4663	0.5374
table-GAN	0.2185	0.4605	0.3482	0.2084
medGAN	0.6807	0.6146	0.3448	0.3050
CTGAN	0.7826	0.6845	0.4561	0.4558
cWGAN	0.7760	0.5253	0.0053	0.2166
CTAB-GAN	0.8138	0.7136	0.4708	0.4911
CasTGAN	0.8048	0.7315	0.2845	0.4995

TABLE 13. Diabetes dataset classification.

	Accuracy	ROCAUC	F1-score	PR-AUC
Identity	0.8623	0.8142	0.2545	0.3980
table-GAN	0.8604	0.5222	0.0000	0.1585
medGAN	0.8519	0.7544	0.2473	0.3241
CTGAN	0.8333	0.7990	0.3924	0.3711
cWGAN	0.8443	0.7636	0.2749	0.3325
CTAB-GAN	0.8432	0.7857	0.3519	0.3589
CasTGAN	0.8557	0.8077	0.3336	0.3866

TABLE 14. Cars dataset regression.

	RMSE	R^2 Score
Identity	4330.3428	0.7604
table-GAN	16737.5294	-2.0148
medGAN	4822.8477	0.7491
CTGAN	6558.5413	0.5327
cWGAN	166818.8351	-319.0963
CasTGAN	6396.3703	0.5566

TABLE 15. Housing dataset regression.

	RMSE	R^2 Score
Identity	393507.3597	-0.3696
table-GAN	2943381.0066	-83.1326
medGAN	405452.4482	-0.3845
CTGAN	458694.4080	-0.6533
cWGAN	518963.3807	-1.1054
CTAB-GAN	472485.7269	-0.7284
CasTGAN	419848.1947	-0.4433

D. CASTGAN TRAINING STABILITY

See Figure 7.

REFERENCES

[1] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher, and S. McLachlan, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 3, pp. 230–238, Mar. 2018, doi: 10.1093/jamia/ocx079.

[2] G. Pavlidis, "Financial information in the context of anti-money laundering: Broadening the access of law enforcement and facilitating information exchanges," *J. Money Laundering Control*, vol. 23, no. 2, pp. 369–378, Mar. 2020, doi: 10.1108/jmlc-10-2019-0081.

[3] (2016). *Regulation (Eu) 2016/679 of the European Parliament and of the Council*. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[5] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018, doi: 10.1109/MCI.2018.2840738.

[6] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2019, pp. 703–716., doi: 10.1007/978-3-030-30490-4_56.

[7] M. Mirza and S. Osindero, ''Conditional generative adversarial nets,'' 2014, *arXiv:1411.1784*.

[8] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, ''Adversarial autoencoders,'' 2015, *arXiv:1511.05644*.

[9] M. Esmaeilpour, N. Chaalia, A. Abusitta, F.-X. Devailly, W. Maazoun, and P. Cardinal, ''Bi-discriminator GAN for tabular data synthesis,'' *Pattern Recognit. Lett.*, vol. 159, pp. 204–210, Jul. 2022, doi: 10.1016/j.patrec.2022.05.023.

[10] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, ''Generating synthetic data in finance: Opportunities, challenges and pitfalls,'' in *Proc. 1st ACM Int. Conf. AI Finance*, Oct. 2020, pp. 1–8, doi: 10.1145/3383455.3422554.

[11] A. Odena, C. Olah, and J. Shlens, ''Conditional image synthesis with auxiliary classifier GANs,'' in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

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

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, ''Improved training of Wasserstein GANs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[14] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, ''Membership inference attacks against machine learning models,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18, doi: 10.1109/SP.2017.41.

[15] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, ''Privacy risk in machine learning: Analyzing the connection to overfitting,'' in *Proc. IEEE 31st Comput. Secur. Found. Symp. (CSF)*, Jul. 2018, pp. 268–282, doi: 10.1109/CSF.2018.00027.

[16] C. Dwork, ''Differential privacy: A survey of results,'' in *Proc. Int. Conf. Theory Appl. Models Comput.* Cham, Switzerland: Springer, 2008, pp. 1–19, doi: 10.1007/978-3-540-79228-4_1.

[17] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, ''LOGAN: Membership inference attacks against generative models,'' *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 1, pp. 133–152, Jan. 2019, doi: 10.2478/popets-2019-0008.

[18] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, ''Differentially private generative adversarial network,'' 2018, *arXiv:1802.06739*.

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

[20] J. Chen, W. H. Wang, H. Gao, and X. Shi, ''PAR-GAN: Improving the generalization of generative adversarial networks against membership inference attacks,'' in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2021, pp. 127–137, doi: 10.1145/3447548.3467445.

[21] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, ''PrivBayes: Private data release via Bayesian networks,'' *ACM Trans. Database Syst.*, vol. 42, no. 4, pp. 1–41, Dec. 2017, doi: 10.1145/3134428.

[22] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, ''Generation and evaluation of synthetic patient data,'' *BMC Med. Res. Methodology*, vol. 20, no. 1, pp. 1–40, Dec. 2020, doi: 10.1186/s12874-020-00977-1.

[23] J. P. Reiter, ''Using cart to generate partially synthetic public use microdata,'' *J. Off. Statist.*, vol. 21, no. 3, p. 441, 2005.

[24] D. S. Watson, K. Blesch, J. Kapar, and M. N. Wright, ''Adversarial random forests for density estimation and generative modeling,'' in *Proc. Int. Conf. Artif. Intell. Statist.*, 2023, pp. 5357–5375.

[25] D. P. Kingma and M. Welling, ''Auto-encoding variational Bayes,'' 2013, *arXiv:1312.6114*.

[26] Z. Wan, Y. Zhang, and H. He, ''Variational autoencoder based synthetic data generation for imbalanced learning,'' in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–7, doi: 10.1109/SSCI.2017.8285168.

[27] Z. Islam, M. Abdel-Aty, Q. Cai, and J. Yuan, ''Crash data augmentation using variational autoencoder,'' *Accident Anal. Prevention*, vol. 151, Mar. 2021, Art. no. 105950, doi: 10.1016/j.aap.2020.105950.

[28] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, ''Modeling tabular data using conditional GAN,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[29] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, ''beta-VAE: Learning basic visual concepts with a constrained variational framework,'' in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.

[30] J. Lee, J. Hyeong, J. Jeon, N. Park, and J. Cho, ''Invertible tabular GANs: Killing two birds with one stone for tabular data synthesis,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 4263–4273.

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

[32] J. Lee, M. Kim, Y. Jeong, and Y. Ro, ''Differentially private normalizing flows for synthetic tabular data generation,'' in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 7, pp. 7345–7353, doi: 10.1609/aaai.v36i7.20697.

[33] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, ''Data synthesis based on generative adversarial networks,'' *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, Jun. 2018, doi: 10.14778/3231751.3231757.

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

[35] J. Engelmann and S. Lessmann, ''Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning,'' *Exp. Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114582, doi: 10.1016/j.eswa.2021.114582.

[36] Z. Zhao, A. Kunar, R. Birke, and L. Y. Chen, ''CTAB-GAN: Effective table data synthesizing,'' in *Proc. 13th Asian Conf. Mach. Learn.*, vol. 157, V. N. Balasubramanian and I. Tsang, Eds. Nov. 2021, pp. 97–112.

[37] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, ''Recent progress on generative adversarial networks (GANs): A survey,'' *IEEE Access*, vol. 7, pp. 36322–36333, 2019, doi: 10.1109/ACCESS.2019.2905015.

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

[39] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, ''VEEGAN: Reducing mode collapse in GANs using implicit variational learning,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[40] E. Strelcenia and S. Prakoonwit, ''A survey on GAN techniques for data augmentation to address the imbalanced data issues in credit card fraud detection,'' *Mach. Learn. Knowl. Extraction*, vol. 5, no. 1, pp. 304–329, Mar. 2023, doi: 10.3390/make5010019.

[41] J. Kim, J. Jeon, J. Lee, J. Hyeong, and N. Park, ''Oct-GAN: Neural ode-based conditional tabular gans,'' in *Proc. Web Conf. 2021*, 2021, pp. 1506–1515, doi: 10.1145/3442381.3449999.

[42] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, ''COVIDGAN: Data augmentation using auxiliary classifier GAN for improved COVID-19 detection,'' *IEEE Access*, vol. 8, pp. 91916–91923, 2020, doi: 10.1109/ACCESS.2020.2994762.

[43] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ''LightGBM: A highly efficient gradient boosting decision tree,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.

[44] D. M. Blei and M. I. Jordan, ''Variational inference for Dirichlet process mixtures,'' *Bayesian Anal.*, vol. 1, no. 1, pp. 121–143, Mar. 2006, doi: 10.1214/06-ba104.

[45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, ''Improved techniques for training GANs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, ''Layer normalization,'' 2016, *arXiv:1607.06450*.

[47] R. Kohavi and B. Becker. (1996). *Adult Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult

[48] S. Moro, P. Cortez, and P. Rita. (2012). *Bank Marketing Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/bank+marketing

[49] I.-C. Yeh and C.-H. Lien. (2016). *Default of Credit Card Clients Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[50] A. Taboul. (2021). *Diabetes Health Indicators Dataset*. [Online]. Available: https://www.kaggle.com/alexteboul/diabetes-health-indicators-dataset

[51] King County. (2021). *House Sales in King County, USA*. [Online]. Available: https://www.kaggle.com/harlfoxem/housesalesprediction

[52] B. Tunguz. (2021). *Used Cars Auction Prices*. [Online]. Available: https://www.kaggle.com/tunguz/used-car-auction-prices/code

[53] I. M. Chakravarti, R. G. Laha, and J. Roy, *Handbook of Methods of Applied Statistics* (Wiley Series in Probability and Mathematical Statistics). Wiley, 1967.

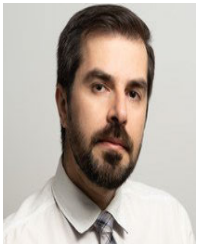
[54] M. Padala, D. Das, and S. Gujar, ''Effect of input noise dimension in gans,'' in *Proc. Int. Conf. Neural Inf. Process.*, 2021, pp. 558–569.

[55] I. Goodfellow, ''NIPS 2016 tutorial: Generative adversarial networks,'' 2017, *arXiv:1701.00160*.

ABDALLAH ALSHANTTI received the B.Eng. degree in mechatronic and robotic engineering from The University of Sheffield, in 2016, and the M.Sc. degree in artificial intelligence from the University of Southampton, in 2017. He is currently pursuing the Ph.D. degree with the Norwegian University of Science and Technology (NTNU). He has been a Data Scientist with DNB ASA, since 2022. His research interests include statistical modeling, machine learning, generative artificial intelligence, and anti-money laundering.

DAMIANO VARAGNOLO received the Dr.Eng. degree in automation engineering and the Ph.D. degree in information engineering from the University of Padova, in 2005 and 2011, respectively. He was a Research Engineer with Tecnogamma S.p.A., Treviso, Italy, from 2006 to 2007, and visited UC Berkeley as a Scholar Researcher, in 2010. From March 2012 to December 2013, he was a Postdoctoral Scholar with the KTH Royal Institute of Technology, Stockholm. From January 2014 to November 2019, he was a Lecturer with the Luleå University of Technology, Sweden, and then he was a Professor with the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, with a joint position with the University of Padova, in 2022. His research interests include statistical learning, distributed optimization, and distributed nonparametric estimation, with a special focus on applications, including identification and control for the built environment, learning analytics, and muscular rehabilitation.

ADIL RASHEED is currently a Professor with the Department of Engineering Cybernetics, Norwegian University of Science and Technology. There, he works to advance the development of novel hybrid methods that combine big data, physics-driven modeling, and data-driven modeling in the context of real-time automation and control. In addition, he is also a part-time Senior Scientist with the Department of Mathematics and Cybernetics, SINTEF Digital, where he was the Leader of the Computational Sciences and Engineering Group, from 2012 to 2018. His contributions in these roles have been the development and advancement of both the hybrid analysis and modeling and big data cybernetics concepts. Over the course of his career, he has been the driving force behind numerous projects focused on different aspects of digital twin technology, ranging from autonomous ships to wind energy, aquaculture, drones, business processes, and indoor farming. He is also leading the digital twin and asset management related work with the FME Northwind Center.



ARIA RAHMATI received the Ph.D. degree from the Norwegian University of Science and Technology (NTNU), in 2016, specializing in the multifaceted realms of computer vision, machine learning, and medical engineering. Following the successful completion of his doctoral journey and making a meaningful impact in academia through a series of articles, driven by passion for practical applications, he transitioned into the industry, where he embarked on the realm of data both as a Data Scientist and a Data Engineer. He continues to be a driving force in bridging the gap between research and real-world solutions.

FRANK WESTAD is currently an Adjunct Professor with the Department of Engineering Cybernetics, Norwegian University of Science and Technology. There, he lectures in multivariate data analysis, machine learning, and design of experiments. In addition, he is the Lead Data Scientist with Idletechs AS, a company working with thermal imaging in the metallurgy industry. He has experience from numerous industries, such as pharma, food, and chemistry, mainly for process modeling with multichannel sensors for real-time process monitoring and prediction.