HydraGAN – A Multi-head, Multi-objective Approach to Synthetic Data Generation

Chance DeSmet, Diane J. Cook

November 16, 2021

Abstract

Synthetic data generation overcomes limitations of real-world machine learning. Traditional methods are valuable for augmenting costly datasets but only optimize one criterion: realism. In this paper, we tackle the problem of generating synthetic data that optimize multiple criteria. This goal is necessary when real data are replaced by synthetic for privacy preservation. We introduce HydraGAN, a new approach to synthetic data generation that introduces multiple generator and discriminator agents into the system. The multi-agent GAN optimizes the goal of privacy-preservation as well as data realism. To facilitate multi-agent training, we adapt game-theoretic principles to offer equilibrium guarantees. We observe that HydraGAN outperforms baseline methods for three datasets for multiple criteria of maximizing data realism, maximizing model accuracy, and minimizing re-identification risk.

Introduction

The growing availability of personal computational resources has resulted in an explosion in the amount of personal data that are collected and, occasionally, disseminated. From banking details to recorded preferences and detailed behavior sensor readings, IoT devices have been the harbingers of a massive increase in the amount of insightful and sensitive individualized data. Private data may not need to come from a source that is specifically focused on gathering information. For example, smart grid information may introduce privacy concerns, even though the system does not ostensibly exist to track users [10, 47]. Accompanying this rapid increase in big data production and consumption is the need to protect sensitive content from malicious adversaries. Sharing data can support collaboration and promote new findings and advances, yet sharing sensitive data may violate privacy constraints and even endanger individuals whose personal information can be derived from the data [14, 43].

To ensure that sensitive information is not leaked, many privacy-preserving data mining methods have been suggested in the literature [1, 41, 42, 47]. These methods all face a common privacy/utility trade-off. Specifically, an increase in privacy is accompanied by a commensurate decrease in data realism, and thus model accuracy [12, 28, 37]. In the most extreme case, privacy can be preserved by removing sensitive information or modifying these features beyond recognition. However, each modification may introduce changes and biases in the underlying concepts that can be learned. This negative correlation between data privacy and data utility may also be perceived as two ends of a spectrum. Therefore, the choice of Privacy-Preserving Data Mining (PPDM) method is often one that adapts best to the goals of the analysis task, such as the desired amount of privacy or the required utility of the predictive model.

To optimize the multiple competing goals of data realism, model integrity, and data privacy, we introduce HydraGAN, a multi-agent generative adversarial network (GAN) that game-theoretically optimizes multiple criteria to generate privacy-preserving, realistic synthetic data. We hypothesize that data generated by HydraGAN meet the multiple criteria that are required for privacy-preserving data mining. HydraGAN offers several novel contributions:

1. HydraGAN introduces a privacy-preserving discriminator into a GAN framework.
2. HydraGAN divides the data space among multiple generators to promote data diversity.
3. HydraGAN ensures privacy by adversarially attempting to reidentify sensitive attributes. Instead of using a static preservation method such as noise infusion, our system trains a discriminator to reidentify sensitive attributes from the generated synthetic data. This discriminator dynamically improves its ability through training, simulating an actual “information warfare” attack an adversary would make to access private information. This process conditions the generators to not provide data that are vulnerable to reidentification.
4. HydraGAN is designed as a “multi-headed” GAN that combines any number of generators and discriminators. The method regulates the multi-agent training process using game-theoretic techniques.

In this paper, we introduce the notion of a multi-agent GAN and formally verify that the game-theoretic algorithm will reach an equilibrium between all contributing agents. We discuss how this method can be instantiated to perform privacy-preserving data mining. To evaluate algorithm performance, we compare HydraGAN with other reported methods on real and synthetic datasets using the multiple performance metrics of data utility, data privacy, and data realism.

Related Work

Privacy Preserving Data Mining

The goal of privacy preserving data mining (PPDM) is to analyze sensitive data in a manner that does not reveal sensitive information. PPDM techniques range from removing the sensitive information \[8\] or adding noise to the data before analysis \[34\] to generating entirely new data based upon the original \[26\]. HydraGAN fits into this last category by generating synthetic data that both retain the privacy of sensitive information and are faithful to the original data.

Differential Privacy

Differential privacy is an increasingly common PPDM metric used to measure if the inclusion or removal of an individual’s information would incur a privacy loss greater than a given bound, \(\epsilon\) \[13, 18, 23\]. The metric offers quantitative bounds and adapts to varied use cases \[5, 9, 19, 27\]. Since its inception in 2006 \[13\], differential privacy has rapidly increased in popularity and is now considered the ‘de facto’ privacy metric across a broad spectrum of domains \[45\]. Liu et al. \[29\] explore the use of differential privacy to quantify preservation of user identifiers in an eye-tracking study. Their PPDM method added noise to the amount of time a person spent looking at each pixel on ‘gaze maps.’ The amount of noise added to each gaze map ensured that adversaries would not be able to infer the form of a given individual’s gaze map, even if an aggregated noise map was collected and the adversary held prior knowledge of other individuals. Differential privacy may also be extended to meet increased privacy requirements. For example, local differential privacy provides a stronger guarantee than standard differential privacy, as it necessitates that an individual’s data must be privatized before it is collected. To scale their work, Cormode et al. \[11\] created a distributed version of local differential privacy. This extension afforded individuals the privacy guarantees of local differential privacy under more specific, and difficult to privatize, queries.

Reidentification Risk

Because PPDM secures sensitive information, assessing effectiveness equates to measuring the risk of reidentifying sensitive information. A primary type of reidentification attack is known as a ‘linkage attack.’ This type of attack attempts to combine multiple records for a single person to piece together sensitive information. As an example, if an adversary had access to an individual’s work schedule, along with their electric bill, the adversary could combine these insights to deduce when the person was away from their home and could subsequently cause harm to the person. Over time, linkage attacks have become more sophisticated, using techniques such as probabilistic means to establish likely correlations between users \[6, 31, 39\]. HydraGAN protects against these linkage attacks by removing the original person from the data. Any ‘individual’ present in the data generated by HydraGAN is in reality an amalgamation of the data from many different persons that trained the model, with additional privacy provided through its active reidentification discriminator. Because of this process, HydraGAN-generated data would be difficult to use in a linkage attack.

Generative Adversarial Networks

In recent years, Generative Adversarial Networks have generated increasingly realistic synthetic data, evolving from relatively simple tasks to complex, multi-target generation. Introduced by Goodfellow et al. \[16\], a GAN typically consists of two networks, a generator that proposes specific data configurations, and a discriminator that attempts to
differentiate generated data from real samples. As the generator learns how better to fool the discriminator, the discriminator correspondingly improves its perceptiveness at discovering subtle deviations from the original data. Recent works have updated GAN design to include multiple generator or discriminator networks. For instance, CycleGAN [48], uses two discriminators and two generative mappings to learn how to translate images between different domains. Similarly, Hardy et al. introduced MDGAN [17], that extends federated learning to a series of discriminators that represented by multiple distributed system. This method allowed a single, central generator to efficiently learn from multiple distributed systems. These GAN innovations set the stage for HydraGAN, a method that attempts to take advantage of the flexibility of multi-agent GANs to generate synthetic, private data.

Synthetic Data Generation

HydraGAN’s ensures the privacy of sensitive data by generating synthetic data that obfuscates sensitive information while retaining the predictive concepts within the original data. Synthetic data generation techniques vary widely in form, including probabilistic approaches to simulating epidemiological information [15][20][33], deep methods such as autoencoders that create synthetic medical images [40][44][46], and Markov models being that synthesize and predict network traffic [25].

Due to their strength in generating realistic data, GANs have recently become the de facto method for creating synthetic data. GAN-based synthetic data generation is primarily limited to image data. One influential work is med-GAN which translates medical images into new domains, assisting clinicians [3]. Despite the resourceful data generation design, there is a potential for an adversary to discover the training examples used in the creation of synthetic data. To combat this attack, a new class of generator was introduced that attempts to secure the original data from malicious attack. Two examples are PATE-GAN [22] and PPGAN [30]. These methods, like HydraGAN, build on a Generative Adversarial Network (GAN) structure to generate privacy-preserving synthetic data. PATE-GAN provides privacy by partitioning the original data set, training a set of discriminators on different data partitions, then noisily aggregating the results to be able to offer differential privacy guarantees to the resultant data [22]. PPGAN instead provides privacy by adding noise to the gradients of the discriminator, thus also offering a demonstrable amount of differential privacy to the generated data [30].

In contrast to these approaches, the goal of the HydraGAN algorithm is not to achieve differential privacy by adding uncertainty to the training process. We do not build on such prior work because adding noise may degrade the performance of models that rely on data quality. We believe that dynamically learning to safeguard against possible attacks may be more protective than static, one-time noise addition methods. To this end, HydraGAN learns to safeguard the data much the same way that a standard GAN learns to generate real data: by letting the generator iteratively compete against an adversary that wants to compromise the data.

Framework

HydraGAN builds upon the Wasserstein GAN (WGAN) [2]. Like a WGAN, HydraGAN contains a generator/discriminator pair in which the discriminator grades the realism of samples produced by the generator. From this starting point, we propose several new design elements to facilitate the optimization of multiple criteria. In this paper, we focus on the criteria of data realism and privacy preservation. This goal is accomplished through several changes to the traditional structure: the inclusion of a multi-headed generator, a re-identification discriminator, and a game-theoretic loss function to enable these components to interact appropriately.

Figure 1 illustrates the HydraGAN architecture. To prevent mode collapse, the generator contains multiple “heads”, or generator functions, that operate on independent subsets of the real data. The multi-headed generator initially processes a noise vector and produces sets of synthetic samples, one set per generator head. Each set of samples is evaluated by a pair of discriminators, tasked with evaluating realism and re-identification. The discriminators process samples from each generator head separately and score each generated output independently. After each training iteration, each learning agent (i.e., each generator head and each discriminator) is updated based on the loss from the previous iteration. This update uses a custom loss function based on Equations 4 through 13 to synchronize training for all the HydraGAN agents.
Multi-head Generator

GANs are plagued by the issue of mode collapse \cite{48}, which increases the difficulty of training the agents. Mode collapse often occurs when either the generator or discriminator diverges from its counterpart, resulting in the GAN producing unrealistic or repetitive synthetic samples \cite{38}. There are several common causes of mode collapse. A generator may produce data that are so similar to the original that the discriminator cannot differentiate between the two, resulting in the generator solely producing samples within this small distribution. On the discriminative side, a discriminator may learn to differentiate between the original and generated data to a degree that the generator is not able to fool the discriminator. In this case, the generator starts creating the same data continuously without any additional learning.

To lessen the impact of mode collapse, we partition the real data into clusters using k-means clustering. We select $k$ using the “elbow” technique \cite{7}, increasing $k$ until the change in performance measure (e.g., the sum of squared distances between points and cluster centroids) converges.

Next, we create a generator with multiple output “heads”. Here, every generative head originates from a shared series of layers. Figure 2 contains the details of the generator construction. This version of the generator contains five heads, each outputting data described by fourteen features. Each head is the same shape, and they all originate from a single layer in the generator, as shown in the figure. Each head is connected to an individual discriminator pair, and every discriminator/head pair is responsible for simulating one of the clusters. This partitioning can also be valuable if different data subsets are accompanied by different privacy needs. Through this construction, mode collapse is addressed in two ways. First, the simplification of the generated data distributions facilitates a more easily learned loss landscape for the generator and discriminator. Second, as each generative head is propagated from a shared set of layers, a generative head suffering from mode collapse may be fixed through updates to these initial layers originating from the other, better performing, heads.
Figure 3: The network structure of the discriminators in HydraGAN.

Realism Discriminator

HydraGAN’s realism discriminator is tasked with gauging how faithfully the generated data emulate the real data. Figure 3 illustrates this discriminator’s structure. The network processes input data with fourteen features, convolves the input, and applies dense layers before a final score is produced. Convolutional layers are common for image GANs. We employ them in HydraGAN as well to maintain relationships between data features. After training, this discriminator outputs a single variable, $r \in [-\infty, \infty]$, indicating the likelihood that the input data are real. As with other WGANs, gradient clipping is added to ensure that the network training occurs within a Lipschitz space [2].

HydraGAN performance varies based on the activation functions that are employed. We observed a faster training time when the activation function did not have a constant maximum, but yielded depreciating returns as the input to this activation increased. As a result, we introduce the activation function shown in Equation 1. This operates like a Leaky ReLU with the nonlinear ‘leaks’ occurring both below and above the range [-1 1].

\[
F(x) = \frac{x}{|x|} \ast \log(|x| + 1)
\]  

(1)

Re-Identification Discriminator

In contrast to the realism discriminator, the re-identification discriminator’s objective is not to discern real data from synthetic. Instead, this discriminator grades the vulnerability of the generated data to re-identification. To accomplish this goal, the discriminator actively attempts to identify the value of a sensitive feature given the non-sensitive features from the same data point. Within HydraGAN, a quadratic loss function is used to incentivize the generator to produce samples that cause the prediction given by the re-identification to be different than what was actually output by the generator. As the re-identification discriminator trains on real samples to learn how to re-identify the sensitive attribute, the generator will have to iteratively produce samples that do not lend themselves to an easy inference of the sensitive attribute (Equation 2). Additionally, in order to improve training speed, we do not activate the re-identification discriminator until the EM distance between the generated and real data is less than 0.3.

\[
F(y_{true}, y_{pred}) = (|y_{true} - y_{pred}| - 0.25)^2
\]  

(2)

Proof of Game-Theoretic Equilibrium

In the traditional GAN, the game-theoretic interplay is evident. The generator and discriminator are bound together in a zero-sum game; thus, one cannot perform better without its counterpart performing worse. The two networks form a minmax ‘game,’ where a Nash equilibrium is reached when both networks are performing close to the same level and both have reached a local (or global) optimal state. When transitioning to a multi-player GAN, this equilibrium is not guaranteed to exist. In practice, this could be due to the generator successfully fooling the re-identifier while performing poorly against the realism discriminator. Formulating the multi-objective equilibrium presents a challenge. When the generator adjusts its model to yield more realistic data, the resulting network may experience deteriorating re-identification performance. However, this trend may reverse on subsequent training rounds. Under these constraints,
there is no guarantee that a convergence will ever occur, as the networks may oscillate between several different poorly-performing states. We change the situation by modifying the WGAN loss function to also include the summed losses of the other HydraGAN agents. This change links the networks into a collaborative partnership: an individual agent cannot perform poorly without its poor performance impacting the training of the other GAN agents.

We formalize our proposed approach by adapting the convergence equation introduced by Kodali et al. [24]. Here, we offer a proof that using the modified HydraGAN loss function, if the generator and a discriminator reach an equilibrium, that point also represents an equilibrium for the rest of the discriminators.

In this proof, we first define the cost functions for a standard GAN and describe how those cost functions establish an equilibrium between the single generator / discriminator pair. We then introduce our new cost function. Using the new cost function, we verify that when an equilibrium is reached for a single generator-discriminator pair, it follows that an equilibrium is reached for all of the HydraGAN agent pairs.

We first restate the traditional GAN cost function, shown in Equations [3] and [4]. In these equations, \( J_n(\phi, \theta) \) represents the cost function for a network \( n \) with parameters \( \phi \) and \( \theta \). These parameters represent the current weights of the generator and discriminator, respectively. The right hand side of the equations follow the notation introduced by Goodfellow et al. [10] for the generator and discriminator components of the loss function. In these equations, \( E_X \) draws a sample from the real data distribution and \( E_z \) draws a sample from some input noise. Next, \( D_\phi(x) \) represents the performance of the discriminator given samples \( x \) with weights \( \theta \), and \( G_\phi(z) \) represents the generator’s output based on input \( z \) using weights \( \theta \). The goal of the original two-player GAN is to find parameters \( \theta \) that minimize the log probability of \( 1 - D(G(z)) \). The term \( D(G(z)) \) represents the probability that generated data \( G(z) \) is real. If the discriminator correctly classifies a fake input, then \( D(G(z)) = 0 \).

**Proof.**

\[
J_G(\phi, \theta) := E_{X \in \text{real}} \log D_\phi(x) + E_z \log(1 - D_\theta(G_\phi(z))) \tag{3}
\]

\[
J_D(\phi, \theta) := -E_{X \in \text{real}} \log D_\phi(x) - E_z \log(1 - D_\theta(G_\phi(z))) \tag{4}
\]

We first define the cost functions for both the generator and the discriminator in a two-agent system. Equation [4] represents the cost from the perspective of the discriminator, evaluating how well it can identify if the input values \( x \) are drawn from the real data distribution or if they are synthetic examples provided by the generator. Equation [3] describes the cost function in terms of the generator. This cost function considers the generator’s performance in fooling the discriminator into labeling its output as “real”.

Equations [5] and [6] introduce the equilibrium equations from Kodali et al. [24]. These equations use the cost function \( J_n(\phi, \theta) \) to describe an equilibrium between the generator and the discriminator. What these equations show is that for the parameters \( \phi \) and \( \theta \), a change of these parameters within some bound \( \gamma \) will not result in a superior cost function performance for either network. The term \( \gamma \) represents the current parameterization of the network, and \( \ast \) represents a new parameterization within some maximum distance \( \gamma \) away from \( \gamma \). An \( \epsilon \) term is included to provide an error bound. Equations [5] and [6] thus represent an equilibrium such that all of the weights in the system are in a position where any change would result in diminished performance.

\[
\forall \phi', ||\phi' - \phi^\ast|| \leq \gamma : J(\phi^\ast, \theta^\ast) \leq J(\phi', \theta^\ast) + \epsilon \tag{5}
\]

\[
\forall \theta', ||\theta' - \theta^\ast|| \leq \gamma : J(\phi^\ast, \theta^\ast) \geq J(\phi^\ast, \theta') - \epsilon \tag{6}
\]

In HydraGAN, multiple discriminators are present, each with its own objective. To accommodate these various, sometimes competing goals, we introduce a new cost function in Equation [7]. In this example, the cost function references particular discriminator \( j \). Additionally, this function includes the individual cost of the remaining discriminators with the addition of a \( \lambda \) term that sums this cost of all discriminators other than \( j \). In the multi-agent formulation, calculating the loss of an agent must take into account the rest of the HydraGAN discriminators, essentially combining the losses. We will now verify that using this new cost function will allow us to arbitrarily choose an equilibrium for a specific generator/discriminator pair. By rearranging the terms in Equation [7] we will demonstrate that this is an equilibrium point for all discriminators.

\[
J_D(\phi, \theta, \lambda) := E_X \log(D_{\theta_j}(x)) + E_z \log(1 - D_{\theta_j}(G_\phi(z))) \nonumber
\]

\[
+ \sum_{i \in \lambda, i \neq j} E_X \log(D_{\theta_i}(x)) + E_z \log(1 - D_{\theta_i}(G_\phi(z))) \tag{7}
\]

Equation [8] expresses the new equilibrium equation. In this question, \( \lambda \) represents the rest of the cost functions present in HydraGAN.
∀θ', ||θ' - θ*|| ≤ γ : J(ϕ*, θ*, λ) ≥ J(ϕ*, θ', λ) - ε 

(8)

We now label θ in terms with our selected discriminator j, as shown in Equation 9:

∀θ', ||θ' - θ*|| ≤ γ : J(ϕ*, θj*, λ) ≥ J(ϕ*, θj', λ) - ε

(9)

Next, we expand the new cost functions found in Equation 9, allowing us to view the equilibrium as a composite of the inner terms.

\[
\begin{align*}
&= E_X p_j \log(D_{\theta_j'}(x)) + E_z \log(1 - D_{\theta_j'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq j} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) \\
&\geq E_X p_j \log(D_{\theta_j'}(x)) + E_z \log(1 - D_{\theta_j'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq j} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) - \epsilon
\end{align*}
\]

(10)

We arbitrarily choose an additional discriminator k, and express the cost function associated for discriminator k, along with the cost for discriminator j and the summation costs for the remaining discriminators.

\[
\begin{align*}
&= E_X p_j \log(D_{\theta_j'}(x)) + E_z \log(1 - D_{\theta_j'}(G_{\phi'}(z))) + \\
&E_X p_k \log(D_{\theta_k'}(x)) + E_z \log(1 - D_{\theta_k'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq j,k} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq j,k} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) - \epsilon
\end{align*}
\]

(11)

We now integrate the original discriminator j into the λ term representing the rest of the HydraGAN discriminators. This is possible because all discriminators use the same cost function, as well as the same maximum step size. This addition is a critical step, demonstrating that the λ term representing the rest of the discriminator’s loss functions allows us to ensure that any equilibrium point is shared between all discriminators.

\[
\begin{align*}
&= E_X p_k \log(D_{\theta_k'}(x)) + E_z \log(1 - D_{\theta_k'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq k} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) + \\
&\sum_{i \in \lambda, i \neq k} E_X p_i \log(D_{\theta_i'}(x)) + E_z \log(1 - D_{\theta_i'}(G_{\phi'}(z))) - \epsilon
\end{align*}
\]

(12)

Equation 13 is a transformed version of the original cost function replacing discriminator j with k. Following this process, we see that any equilibrium point between the generator and a given discriminator is an equilibrium point for every other discriminator as well. This result motivates the modification of the standard Wasserstein loss function for each discriminator into a loss function that also accounts for the other discriminator in HydraGAN, allowing for a more equitable training process that is not dominated by either the realism or re-identification discriminator.
Experimental Results

Our goal is to create synthetic data that satisfy multiple objectives. In this paper, our objectives are 1) data realism, 2) performance of a predictive model trained on the data, and 3) privacy preservation. In our work, we view privacy preservation as obscuring the values of sensitive features. If accessed either unintentionally or maliciously by an external source, sensitive features may cause damage to one or more parties. To evaluate the utility of HydraGAN-generated data, we compare its performance along these multiple dimensions with baseline synthetic data generation methods. The first of the two hyperparameters tuned during the implementation and testing of this model were the learning rates for both the RMSprop and Adam optimizers, which ranged during testing from \([0.0000001, 0.1]\), with best performance at 0.0002. The second parameter was the clip constraint of the weight clipping, ranging in tests from \([0.001, 0.5]\), with best performance at 0.05.

We validate the methods on three datasets. The first real-world dataset is the UCI Heart Disease dataset \([21]\). In this data, each sample contains 13 physiological characteristics, including an attribute indicating whether the specific individual was diagnosed with heart disease. The second dataset contains 53 features extracted from ambient sensor data readings collected in smart homes \([36]\). Sensor data were collected while subjects performed scripted activities in a smart home testbed. The testbed contains ambient sensors that monitor motion (with a passive infrared motion detector), ambient light and temperature, and door usage (with magnetic door closure sensors). In these two datasets relating to human participants, the sensitive value we will use will is participant age, as previous studies reveal this is a feature that is vulnerable to a re-identification attack \([32]\). The third dataset represents electrical power consumption, reporting load, reaction time, and power balance under various conditions, from a work by Arzamasov et al. \([4]\). We treat the “stability” variable as sensitive for this analysis.

The metrics of realism and privacy often compete; thus, we examine performance separately along these dimensions. To evaluate data realism, we will calculate the inverse of the Earth Mover’s (EM) distance between the original and generated data (Inverse EM) \([35]\). To evaluate a model’s predictive performance trained on the data, we compute average predictive error across three classifiers (support vector, K nearest neighbors, and decision tree regressor) trained on the data and trying to predict the values of non-sensitive features in the real data (Inverse Mean MAE). To evaluate privacy preservation, we compute the average error of the same classifiers in predicting the sensitive attribute (ReID MAE). A generator that performs well on these metrics will produce data that are of a similar form to the original, preserve the predictive utility of the real data, and safeguard the sensitive attribute.

To visualize the performance of these models for multiple objectives, we use a radar plot highlighting the three performance dimensions. The goal is to maximize the value along each dimension. In addition, we will compute the correlation between the sensitive attribute and the remaining features. Paired with a visualization of the generated data, we can observe the impact each feature has on the re-identification of sensitive attributes.

From the radar charts, we see that HydraGAN is more adaptive to multiple objectives. In particular, HydraGAN...
attains greater realism than PATEGAN and provides greater privacy preservation than PPGAN. PPGAN focuses its effort more on realism than privacy preservation, yet HydraGAN’s realism is comparable based on model MAE as well as EM distance, while providing a much greater level of privacy to the sensitive attributes of age (Figures 4 and 5) and power consumption (Figure 6). As would be expected, as the method offers greater privacy, the data become less consistent with the original. This corresponds with the well-known PPDM concept of the utility/privacy trade-off [12], where increasing the amount of privacy provided to participants in some data will always result in a cost to the realism of that data.

Next, we plot statistics of HydraGAN-generated data and the original UCI heart data. As Figure 7 (left) illustrates, the mean and standard deviation of most features align well. To understand why not all features align, we also plot the absolute value of the correlation coefficients between feature (numbered randomly in Figure 7) and the sensitive attribute (feature 0 in the figure). As the figure illustrates, the stronger the correlation between a feature and the sensitive attribute, the greater is the difference between statistics for the original and synthetic feature values. While this difference can be accomplished by infusing noise, the radar charts indicate the dramatic detrimental impact this technique has on data realism and model predictability. Looking at Figure 7 (right), we further note that PPGAN preserves the relationship between the sensitive and non-sensitive features very well, while HydraGAN preserves some relationships and not others. This may be due to HydraGAN’s re-identification discriminator obscuring re-identification of the sensitive feature by blurring the relationships for highly-correlated features.

Conclusions

We introduce a novel multi-objective synthetic data generator, called HydraGAN. This multi-agent GAN balances a need for realistic data with preserving private information. We validated the ability of HydraGAN to achieve a system equilibrium if one exists. Additionally, experiments real-world datasets indicate that HydraGAN can optimize these multiple, competing performance criteria, resulting in synthetic data that are realistic, retain the predictive performance of real data, and maintain the privacy of sensitive attributes. Further work will incorporate and evaluate additional discriminators. Future work will also investigate refinements of the clustering method for promoting data diversity.
Table 1: Comparison of original data and three privacy-preserving methods for the metrics of realism (Inverse EM), model predictability (Inverse Model MAE), and re-identification (ReID MAE).

| Method | Inverse EM | Inverse Model MAE | ReID MAE |
|--------|------------|-------------------|----------|
| Heart  |            |                   |          |
| Original | 1.00 | 0.96 | 0.02 |
| HydraGAN | 0.82 | 0.72 | 0.46 |
| PPGAN | 0.96 | 0.89 | 0.08 |
| PATEGAN | 0.00 | 0.00 | 1.00 |
| Smarthome | | | |
| Original | 1.00 | 0.97 | 0.02 |
| HydraGAN | 0.80 | 0.78 | 0.21 |
| PPGAN | 0.91 | 0.88 | 0.06 |
| PATEGAN | 0.00 | 0.00 | 1.00 |
| Smartgrid | | | |
| Original | 1.00 | 0.91 | 0.09 |
| HydraGAN | 0.83 | 0.77 | 0.46 |
| PPGAN | 0.95 | 0.81 | 0.13 |
| PATEGAN | 0.00 | 0.00 | 1.00 |

References

[1] C. C. Aggarwal and P. S. Yu. A General Survey of Privacy-Preserving Data Mining Models and Algorithms. 2008.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. *International Conference on Machine Learning*, 34(70):1–44, 2017.

[3] K. Armanious, C. Jiang, M. Fischer, T. Küstner, T. Hepp, K. Nikolaou, S. Gatidis, and B. Yang. MedGAN: Medical image translation using GANs. *Computerized Medical Imaging and Graphics*, 79, 2020.

[4] V. Arzamasov, K. Bohm, and P. Jochem. Towards Concise Models of Grid Stability. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018*, (2):0–5, 2018.

[5] H. Asi, J. Duchi, and O. Javidbakht. Element Level Differential Privacy: The Right Granularity of Privacy. *arXiv:1912.04042v1*, pages 1–34, 2019.

[6] K. Benitez and B. Malin. Evaluating re-identification risks with respect to the HIPAA privacy rule. *Journal of the American Medical Informatics Association*, 17(2):169–177, 2010.

[7] P. Bholowalia and A. Kumar. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications*, 105(9):17–24, 2014.

[8] M. A. Chamikara, P. Bertok, D. Liu, S. Camtepe, and I. Khalil. Efficient data perturbation for privacy preserving and accurate data stream mining. *Pervasive and Mobile Computing*, 48:1–19, 2018.

[9] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11476 LNCS:375–403, 2019.

[10] J. X. Chin, G. Giaconi, T. T. De Rubira, D. Gimpuz, and G. Hug. Considering time correlation in the estimation of privacy loss for consumers with smart meters. *20th Power Systems Computation Conference, PSCC 2018*, 2018.
[11] G. Cormode, T. Kulkarni, and D. Srivastava. Answering range queries under local differential privacy. Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 1832–1834, 2019.

[12] R. Dong, L. J. Ratliff, A. A. Cárdenas, H. Ohlsson, and S. S. Sastry. Quantifying the Utility–Privacy Tradeoff in the Internet of Things. ACM Transactions on Cyber-Physical Systems, 2(2):1–28, 2018.

[13] C. Dwork. Differential privacy. In 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006), 2006.

[14] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin. A systematic review of re-identification attacks on health data. PLoS ONE, 6(12), 2011.

[15] J. Eno and C. W. Thompson. Generating synthetic data to match data mining patterns. IEEE Internet Computing, 12(3):78–82, 2008.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. NIPS, pages 1–9, 2014.

[17] C. Hardy, E. L. Merrer, and B. Sericola. MD-GAN : Multi-Discriminator Generative Adversarial Networks for Distributed Datasets. 33rd IEEE International Parallel and Distributed Processing Symposium, pages 1–12, 2019.

[18] M. Hilton. Differential Privacy: A Historical Survey. Cal Poly State University, 2018.

[19] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong. DP-ADMM: ADMM-Based Distributed Learning with Differential Privacy. IEEE Transactions on Information Forensics and Security, 15:1002–1012, 2020.

[20] J. Jalkó, E. Lagerspetz, J. Haukka, S. Tarkoma, S. Kaski, and A. Honkela. Privacy-preserving data sharing via probabilistic modelling. pages 1–7, 2019.

[21] A. Janosi, W. Steibrunn, M. Pfisterer, and R. Detrano. Heart Disease Data Set. Technical report, Hungarian Institute of Cardiology, University Hospital, Zurich, Switzerland, University Hospital, Zurich, Switzerland, V.A. Medical Center, Long Beach and Cleveland Clinic Foundation, 1988.

[22] J. Jordan, J. Yoon, and M. Van Der Schaar. PATE-GaN: Generating synthetic data with differential privacy guarantees. 7th International Conference on Learning Representations, ICLR 2019, pages 1–21, 2019.

[23] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. page 193, 2011.

[24] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. On Convergence and Stability of GANs -ICLR18 - Rejected. Iclr2018, pages 1–18, 2017.

[25] S. Kordnoori, H. Mostafaei, S. Kordnoori, and M. Ostadrahimi. Testing the Semi Markov Model Using Monte Carlo Simulation Method for Predicting the Network Traffic. Pakistan Journal of Statistics and Operation Research, 16(4):713–720, 2020.

[26] C. F. Kurz, M. Rehm, R. Holle, C. Teuner, M. Laxy, and L. Schwarzkopf. The effect of bariatric surgery on health care costs: A synthetic control approach using Bayesian structural time series. Health Economics (United Kingdom), 28(11):1293–1307, 2019.

[27] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. Proceedings - IEEE Symposium on Security and Privacy, 2019-May:656–672, 2019.

[28] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. KDD, page 517, 2009.

[29] A. Liu, L. Xia, A. Duchowski, R. Bailey, K. Holmqvist, and E. Jain. Differential privacy for eye-tracking data. Eye Tracking Research and Applications Symposium (ETRA), 2019.

[30] Y. Liu, J. Peng, J. J. Yu, and Y. Wu. PPGAN: Privacy-Preserving Generative Adversarial Network. IEEE ICPADS, (201910212133):985–989, 2020.

[31] B. Malin. Re-identification of familial database records. AMIA Annual Symposium Proceedings, pages 524–528, 2006.
[32] L. Na, C. Yang, C. C. Lo, F. Zhao, Y. Fukuoka, and A. Aswani. Feasibility of Reidentifying Individuals in Large National Physical Activity Data Sets from Which Protected Health Information Has Been Removed with Use of Machine Learning. *JAMA Network Open*, 1(8):1–13, 2018.

[33] J. Reiter. Satisfying disclosure restrictions with synthetic data sets. *Journal of Official Statistics-Stockholm-*, 18(4):1–19, 2002.

[34] M. Rodriguez-Garcia, M. Batet, and D. Sánchez. A semantic framework for noise addition with nominal data. *Knowledge-Based Systems*, 122:103–118, 2017.

[35] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover’s Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[36] M. Schmitter-Edgecombe, C. McAlister, and A. Weakley. Naturalistic Assessment of Everyday Functioning in Individuals with Mild Cognitive Impairment: The Day Out Task. *Neuropsychology*, 26(5):1–21, 2012.

[37] J. Snoke, G. M. Raab, B. Nowok, C. Dibben, and A. Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society. Series A: Statistics in Society*, 181(3):663–688, 2018.

[38] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. *31st Conference on Neural Information Processing Systems*, 2017.

[39] A. B. Sunteb and I. P. Fellegi. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[40] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training Deep Networks With Synthetic Data: Bridging the Reality Gap by Domain Randomization. page 9.

[41] I. Wagner and D. Eckhoff. Technical Privacy Metrics. *ACM Computing Surveys*, 51(3):1–38, 2018.

[42] P. Wang, T. Chen, and Z. Wang. Research on Privacy Preserving Data Mining. *Journal of Information Hiding and Privacy Protection*, 1(2):61–68, 2019.

[43] W. E. Winkler and D. C. Washington. Re-identification Methods for Masked Microdata. Technical report, 2004.

[44] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially Private Generative Adversarial Network. *ACM*, 2018.

[45] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K.-Y. Lam. Local Differential Privacy and Its Applications: A Comprehensive Survey. 14(8):1–25, 2020.

[46] X. Zhang, Y. Fu, S. Jiang, X. Xue, Y. G. Jiang, and G. Agam. Stacked multichannel autoencoder – an efficient way of learning from synthetic data. *Multimedia Tools and Applications*, 77(20):26563–26580, 2018.

[47] F. Zhou, J. Anderson, and S. H. Low. Differential privacy of aggregated dc optimal power flow data. *Proceedings of the American Control Conference*, 2019-July:1307–1314, 2019.

[48] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2242–2251, 2017.