2021 International Conference on Computer Communications and Networks (ICCCN) | 978-1-6654-1278-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCN52240.2021.9522203

Synthetic and Private Smart Health Care Data Generation using GANs

Sana Imtiaz[*‡], Muhammad Arsalan[†], Vladimir Vlassov[*], Ramin Sadre[‡]
[*]KTH Royal Institute of Technology, Stockholm, Sweden
[†]Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
[‡]Université catholique de Louvain, Louvain-la-Neuve, Belgium
{sanaim,vladv}@kth.se, muhammad.arsalan@ovgu.de, ramin.sadre@uclouavin.be

***Abstract*—With the rapid advancements in machine learning, the health care paradigm is shifting from treatment towards prevention. The smart health care industry relies on the availability of large-scale health datasets in order to benefit from machine learning-based services. As a consequence, preserving the individuals' privacy becomes vital for sharing sensitive personal information. Synthetic datasets with generative models are considered to be one of the most promising solutions for privacy-preserving data sharing. Among the generative models, generative adversarial networks (GANs) have emerged as the most impressive models for synthetic data generation in recent times. However, smart health care data is attributed with unique challenges such as volume, velocity, and various data types and distributions. We propose a GAN coupled with differential privacy mechanisms for generating a realistic and private smart health care dataset. The proposed approach is not only able to generate realistic synthetic data samples but also the differentially private data samples under different settings: learning from a noisy distribution or noising the learned distribution. We tested and evaluated our proposed approach using a real-world Fitbit dataset. Our results indicate that our proposed approach is able to generate quality synthetic and differentially private dataset that preserves the statistical properties of the original dataset.**

***Index Terms*—Generative adversarial networks, differential privacy, synthetic data generation, smart health care, fitness trackers.**

I. Introduction

The Internet of Things (IoT) paradigm as we know it today is a fruition of the technological advancements in the area of computer networks and communication, that ensure the functionality of these services driven by highly interconnected components. The mass adoption of IoT devices and services creates a plethora of valuable data pools that have applications in areas such as smart health care [1], smart cities [2], smart farming [3] and personalized medicine [4]. These applications are often driven by machine learning (ML) algorithms which ensure the provision of continuously improving personalized services. However, ML-based algorithms and services require access to huge amounts of sensitive and private data, which might not always be reasonable and in some cases, impossible to obtain and share due to local data protection laws. In particular, the advancements in the health care sector are hindered due to the curse of limited data access.

Data access is limited mainly because of the presence of highly sensitive medical information that not only arises concerns for personal privacy but also the threat of misuse or re-identification. Data protection laws like the EU's General Data Protection Regulation (GDPR) ensure higher public trust in data sharing, and informed use of collected user data by the companies. Realistic synthetic datasets offer the benefits of a) enhanced user privacy with reduced risk of re-identification, b) reduced risk of exposure due to privacy-breaching attacks on ML models such as *model inversion* [5], [6], and c) removal of data that could potentially expose competitive advantage for the data providers; all while maintaining fidelity to the real-world data. Therefore, realistic synthetically generated datasets are poised to accelerate the technological advancements in ML, as these datasets do not suffer from the curse of limited availability and can facilitate wide-scale data sharing and usage by industry and researchers without privacy concerns [7]–[12].

Generative modeling is a popular way to model synthetic datasets. These models learn the probability distributions of the given data and are capable of generating very realistic sample distributions from the same data. Hence, generative models are commonly employed for synthetic data generation as well as data augmentation. GANs [13] and their variants have recently become a widely adopted approach for synthetic dataset generation [10], [14]–[18]. However, generating tabular data with GANs, particularly smart health care data, poses unique challenges [14]. The first challenge is the presence of mixed data types, as the real-world data contains both discrete and continuous variables. The second challenge is to accommodate static and behavioral data types. For example age, height and weight are considered static variables as compared to the activity data, as the latter has a higher frequency of recorded changes in observation. Moreover, the data distributions might not always be Gaussian, which makes them harder to normalize or model with GANs. Finally, the major challenge in real-world data comes from highly imbalanced categorical data, as the individuals may possess widely diverse categorical attributes. Moreover, the frequency of logged measurements differs from individual to individual.

GANs can also be combined with different privacy preservation solutions to ensure strong user privacy in the synthesized datasets. Differential privacy (DP) is one of the most popular

This work is partly funded by the Erasmus Mundus Joint Doctorate program in Distributed Computing (EACEA of the European Commission under FPA 2012-0030).

978-1-6654-1278-0/21/$31.00 ©2021 IEEE

solutions used in combination with GAN. This approach relies on noise addition to either the learning mechanism or directly to the data. Research shows several variants of differentially private GANs that employ the noisy learning mechanism [10], [14], [18]–[23]. These DP-strategies are also applied in combination with different variants of GANs depending on the use cases. Moreover, GANs are being extensively used to generate Electronic Health Records (EHRs) [7], [10], [22], [24]. Esteban et al. [22] use a Recurrent Conditional GAN (RCGAN) to generate synthetic time-series EHRs with the noisy learning process. Similarly, Baowaly et al. [10] generate EHRs by using Wasserstein GANs with gradient penalty (WGANs) and boundary-seeking GANs (BGANs). Their evaluations show BGANs to be more suitable for EHR generation.

We address the problem of generating smart health care records using BGANs. Our smart health care dataset is not only more diverse in nature but also possesses the 3 V's of big data (*volume*, *velocity*, and *variety*) as compared to the EHRs. We also used WGANs in our initial set of experiments in comparison with BGANs. Our findings suggest that BGANs are more suitable for synthetic smart health care dataset generation, due to the faster convergence of the BGANs and higher quality of the generated dataset. Moreover, our proposed approach provides additional privacy preservation by integrating DP in different settings. Our results show that the proposed approach is able to generate realistic smart health care data samples with user privacy guarantees.

Our contributions can be summarized as follows:

- We collect and refine a real-world smart health care dataset from geographically distributed users.
- We augment the collected smart health care dataset to represent diverse nutritional and activity patterns based on age, ethnicity, geolocation, dietary preferences, and other factors.
- We propose a method based on GANs for generating synthetic and tabular time-series data, containing categorical and numerical values, as well as the methods to generate the privacy-preserving versions of the synthesized data samples.
- We have created realistic synthetic and privacy-preserving smart health care datasets with fine-grained nutritional and activity user profiles for open use in research.

II. Preliminaries

A. Generative adversarial networks

GAN [13] is a unique kind of generative architecture that is inspired by the zero-sum game in game theory. It consists of two deep learning models, a generator and a discriminator, trained against each other as shown in Fig. 1. The goal of the generator is to capture or learn the distribution of the actual data and generate new data samples. The discriminator aims to detect whether the data is coming from actual data distribution or is it a fake one generated by the generator, hence acting as a binary classifier. The two models compete with each other to improve their performance until they reach a Nash equilibrium where the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.

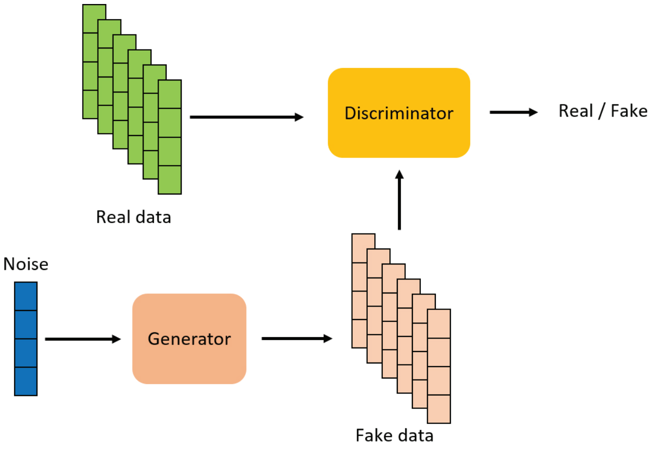


Fig. 1: GAN model for synthetic data generation.

The GAN works as follows: Let G and D be differentiable functions that represent the generator and the discriminator respectively. G takes random variable $\mathbf{z}$ as input and generates a data record $G(\mathbf{z})$ and learns the distribution p_g over data $\mathbf{x}$ with a prior on input noise variables $p_{\mathbf{z}}(\mathbf{z})$. The generated record is then fed to D which also receives the real data record $\mathbf{x}$ from real data distribution $p_{data}(\mathbf{x})$ and tests for their authenticity. The discriminator, when shown both the $\mathbf{x}$ and $G(\mathbf{z})$ data, assigns probabilities $D(\mathbf{x})$ to the record where 1 represents a prediction of the record coming from the real data distribution and 0 represents the data as fake. With time, the discriminator D is trained to maximize the probability of assigning correct labels to both the training examples and the samples generated by G. G is trained simultaneously to minimize the $log(1 - D(G(\mathbf{z})))$. In short, D and G play a two-player minimax with value function $V(G, D)$ given by [13]:

$$\min_G \max_D V(D,G) = E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[logD(\mathbf{x})] + E_{\mathbf{z}} \sim p_{\mathbf{z}(\mathbf{z})}[log(1 - D(G(\mathbf{z})))]. \quad (1)$$

As shown in [13], the optimal discriminator $D^*_G(\mathbf{x})$ is given by:

$$D^*_G(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}. \quad (2)$$

B. Boundary-seeking GAN

Rearranging the equation (2) we get:

$$p_{data}(\mathbf{x}) = p_g(\mathbf{x})\frac{D^*_G(\mathbf{x})}{1 - D^*_G(\mathbf{x})}. \quad (3)$$

From the above equation we can see that even if G is not optimal, the true data distribution can still be found by scaling $p_g(\mathbf{x})$. Furthermore, the optimal generator $p_{data}(\mathbf{x}) = p_g(\mathbf{x})$ can also be obtained by making the discriminator ratio equal to 1, which means that $D(\mathbf{x})$ must be equal to 0.5 and then $D(\mathbf{x}) = 0.5$ is nothing but the decision boundary. For a perfect G, $D(\mathbf{x})$ cannot differentiate between real and fake data, or the real and the fake data are equally likely. Since $D(\mathbf{x})$ has two outputs, each with probability of 0.5, the objective function of G can be modified to force the discriminator outputting 0.5 for every generated data. This can be achieved by minimizing the distance between $D(\mathbf{x})$ and $1 - D(\mathbf{x})$ for all $\mathbf{x}$. Since $D(\mathbf{x})$ is the probability function, the minimum will be achieved at $D(\mathbf{x}) = 1 - D(\mathbf{x}) = 0.5$ and hence the generator G loss is given as [15]:

$$\min_{G} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[\frac{1}{2} (\log D(\mathbf{x}) - \log(1 - D(\mathbf{x})))^2 \right]. \qquad (4)$$

We use the Boundary-seeking GAN in our approach as it offers stable and efficient training.

C. Differential privacy

Differential privacy is a privacy preservation mechanism that aims to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying the underlying records. Privacy is preserved by the addition of noise to either the output of the statistical queries or the input data. DP is provable and quantified using a privacy-loss budget, ε. The most popular mathematical tool used to express DP is as follows [25]:

A randomized algorithm A is ε-differentially private, if for all subsets $S \subseteq Range(A)$ and for all the datasets χ and χ' that differ on at most one row (i.e. the data of one individual),

$$Pr[A(\chi) \in S] \leq e^{\varepsilon} Pr[A(\chi') \in S]. \qquad (5)$$

Here, the ε parameter quantifies the loss of privacy. Absolute privacy is obtained when $\varepsilon = 0$, where the inclusion of one data record has almost no impact on the output of the randomized algorithm A. Achieving high levels of privacy preservation (small ε) requires a higher amount of calculated noise addition which in turn decreases the accuracy of the algorithm [26]. Therefore, a trade-off must be made between keeping the data as private as possible and achieving meaningful and accurate results.

D. The DP post-processing theorem

The post-processing theorem in DP [25] states:

If a mechanism M satisfies ε-DP, and g be any function, then $g(M(x))$ also satisfies ε-DP.

In general, differentially private synthetic data generation algorithms leverage this theorem as they mostly focus on perturbing the distribution. This perturbation is done either directly on the data (by noise addition) or through the learning parameters (noisy learning). As a result, the synthetic data is sampled from a noisy distribution. Applying the post-processing theorem, any data drawn from a noisy distribution (either by parameters or noise addition mechanism) that satisfies DP will also be ε-DP. We employ the noise addition approach as it has been shown that noisy learning may cause GANs to take longer to converge, or in worse cases, not converge at all [27], [28].

E. Laplacian noise addition mechanism

The Laplacian mechanism is one of the most popular noise addition mechanisms in DP [29]. A standard approach is adding random noise with the Laplacian distribution proportional to the sensitivity of the query function S_f to ensure DP-queries. We use the Laplacian differential privacy by adding noise directly to the aggregated data records. Traditionally, for the Laplace mechanism, random noise is drawn from a Laplacian distribution with mean 0 and variance S_f/ε to achieve ε-differential privacy [25]. All the data points in an aggregated data record are individually noised as we pick random noise samples for each point from a $Lap(0, 1/\varepsilon)$ distribution.

III. Data processing pipeline

This section presents our method for data collection, imputation, and transformation. Moreover, we present our approaches for privacy-preserving model training, followed by the data inverse-transformation mechanism. The proposed pipeline is shown in Figure 2. Below, we describe each pipeline stage.

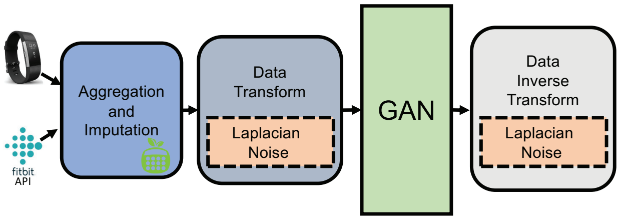


Fig. 2: Data processing pipeline.

A. Data collection and imputation

For this work, we used Fitbit Charge 2 HR smartwatches for automated data collection in combination with the Fitbit App for manually logging user meals. A total number of 25 subjects were observed during this study, distributed across Belgium and Sweden. 12 devices were used for dataset collection, with 2 continual participants (male and female), and 10 users in circulation. The users were asked to record a minimum of 60 days of observations. The participants' pool consisted of 6 coarsely defined ethnicities to represent the overall health and diet patterns of the residing communities. We collected more than $17M$ measurements related to the users' meal logs, calorie intake, heart rate, calories burned, steps taken, activity profile during the day, and sleep. Apart from the numeric data collection, the platform also collects categorical user data, such as age, gender, height, and weight. Since the users were not provided with smart scales, the weight measurement is logged manually. All these logs and measurements were then exported from the Fitbit platform.

1) Time-series data aggregation and imputation: Since this data collection is time-series in nature, with the users logging some manual information, there are natural gaps in the time-series caused by these factors: 1) users forgetting to wear the device or wearing incorrectly, 2) users forgetting to log the meals, and 3) meals not present in food database or customized meals with unavailable nutritional breakdown.

For the gaps in time-series, the days without any meal log entries were omitted. The remaining entries were analyzed for correctness in the recorded measurements. In case of a missing activity profile or a mismatch between the burned calories versus activity profile during that particular day, the user behavior pattern was analyzed to find the closest matching activity profile or the burned calories recorded in the past. A similar approach was used to remove the mismatch between the recorded resting heart rate (RHR) and the steps taken versus the activity profile.

2) Meal logs imputation: Imputation for missing nutritional breakdown for meals is more complex as compared to other missing attributes. When it comes to the available food databases from Fitbit, the United States (US) database is the most largely populated but specialized to the foods available in the US region. On the other hand, the Belgian (French) food database is partially populated. However, since there is no specialized database available for the users in Sweden, the users either recorded the closest matching entry in the US food database or in some cases, the users manually logged the nutritional breakdown for customized meals. A translation API was used to convert logs from other languages to English, replacing the log with either the closest match in the US food database or by using an external nutrition API. Nutritionix API [30] was used to impute the missing nutritional breakdown for meals. Initially, the measurements were aggregated into 3 records. Each contained the nutritional breakdown for a meal (breakfast/lunch/dinner), calories burned during the mealtime, RHR from the previous day, and steps taken as well as the activity records for that day. These records were later aggregated to form one record per day for the nutritional breakdown of all meals, activity profile, steps taken, calories burned, and RHR. The users exhibited all kinds of natural behavior, ranging from very sedentary to highly active users. The complete spectrum of data features (and ranges) is shown in Table I.

Features	**Type**	**Unit**	**Range**
Age	static	yrs	median: 28
Gender	static	-	0: male, 1: female
Height	static	cms	*private*
Weight	static	kgs	*private*
Fat	behavioural	gm	0.08 – 90
Fiber	behavioural	gm	0.06 – 34
Carbs	behavioural	gm	0.06 – 150
Sodium	behavioural	mg	1.92 – 2745
Protein	behavioural	gm	0.14 –75
Calories_burned	behavioural	kcal	1025 – 4331
Resting_heart_rate	behavioural	bpm	49 – 83
Lightly_active_minutes	behavioural	mins	2 – 481
Moderately_active_minutes	behavioural	mins	0 – 211
Very_active_minutes	behavioural	mins	0 – 253
Sedentary_minutes	behavioural	mins	254 – 999
Steps	behavioural	-	162 – 32871

TABLE I: Dataset features with ranges (aggregated per day).

B. Data Transformation

For preparing the data for training, we first remove the *Date* and *Gender* information. Next, the remaining features are normalized and fed to the model for training. Depending on the selected privacy setting (noisy input), we can provide DP-input data to the GAN, which will enable the generation of DP-synthetic samples, as stipulated by the post-processing theorem. Since the data contains categorical or static attributes, which require higher privacy settings, we add Laplacian noise with $\varepsilon = 0.2$ to ensure high noise addition and consequently, stronger privacy settings. On the other hand, behavioral attributes possess a lower risk of re-identification. So we add Laplacian noise with $\varepsilon = 0.5$ to ensure sufficiently high noise addition without losing data utility.

C. Model Training

As mentioned earlier in the Section II-B, we use BGAN for synthetic data sample generation. We trained the BGAN by sampling the population based on gender and geographical location. Moreover, we trained the model in three different privacy settings (non-DP, noisy input, and noisy output) as will be briefly explained in Sec. V-A.

D. Data Inverse-transformation

Once the training is complete and the data is generated by the generator, we first de-normalize the features to better reflect the original data ranges. Afterwards, the columns *Date* and *Gender* are appended to the generated data to make a complete record. Moreover, depending on the chosen privacy setting (noisy output), we add Laplacian noise with $\varepsilon = 0.2$ to the static and with $\varepsilon = 0.5$ to the behavioral variables.

IV. Proposed GAN network for synthetic smart data generation

A. Generator Network

The generator network is shown in Fig. 3. The network takes an input of 15×1 signal followed by 2 dense layers with 64 and 32 neurons, appended with a Leaky ReLU activation with a rate of 0.2. The last dense layer acts as the output layer which takes *tanh* as an activation function.

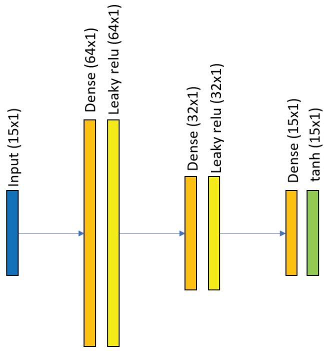


Fig. 3: Generator network.

B. Discriminator Network

The discriminator network is shown in Fig. 4. The network takes an input of 15×1 signal followed by 2 dense layers with 512 and 256 neurons. Both the layers take Leaky ReLU as an activation function with a rate of 0.2. The last layer of the Discriminator network is a dense layer with 1 output and applies *sigmoid* as an activation function.

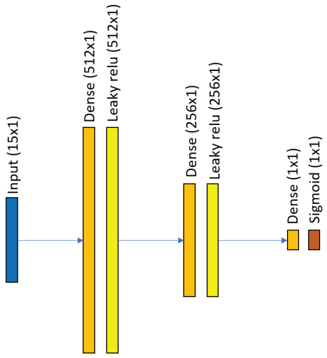


Fig. 4: Discriminator network.

C. Learning rule

We use adaptive moment estimation (Adam) optimizer for both the Discriminator Network and the final GAN network, which computes the adaptive learning rate for each network weight over the learning process from estimates of first and second moments of the gradients. For the configuration parameters, the learning rate α is set to 0.0002 and the exponential decay rate for the first β_1 and second β_2 moment estimates are set to 0.5 and 0.999 respectively. The epsilon that counters the divide by zero problem is set to $1e-8$.

V. Experiments, Results and discussion

We now present our experiments with different additional privacy settings, the respectively generated samples and their histogram distributions.

A. Experiments

As shown in Fig. 2, our pipeline offers multiple points for Laplacian noise addition for differential privacy, enabling 3 experimental settings. The GAN network is able to learn the distribution and to produce plausible examples in each case.

1) Synthetic Data Generation with no Noise Addition: In this setup, the original data is taken as input by the GAN network and the aim is to generate plausible results close to the original data (non-DP).

2) Synthetic Differentially Private Data Generation: In this setup, the network is trained on differentially private data i.e., DP is applied prior to sending it to the discriminator (noisy input). This allows the GAN model to generate differentially private synthetic samples. This approach may be beneficial for settings where synthetic data generation is offloaded to a third party, or when the threat model includes the server node.

3) Applying Differential Privacy to the Synthetic Data: In this setup, we apply DP to the generated synthetic data to observe the effect of noise addition on the quality of generated data (noisy output). This approach offers the advantage in terms of control on the noise addition in generated data, depending on the sensitivity of the data features. However, it requires the additional computation of noise that is added to each generated data point individually.

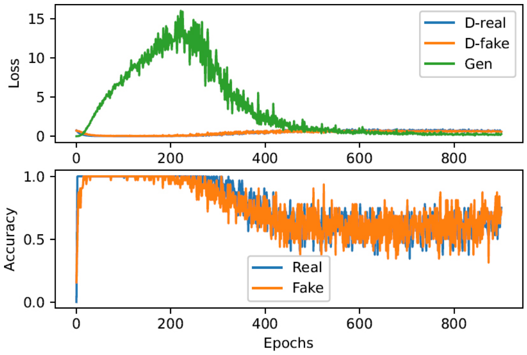


Fig. 5: Line Plots of Loss and Accuracy for a Stable GAN.

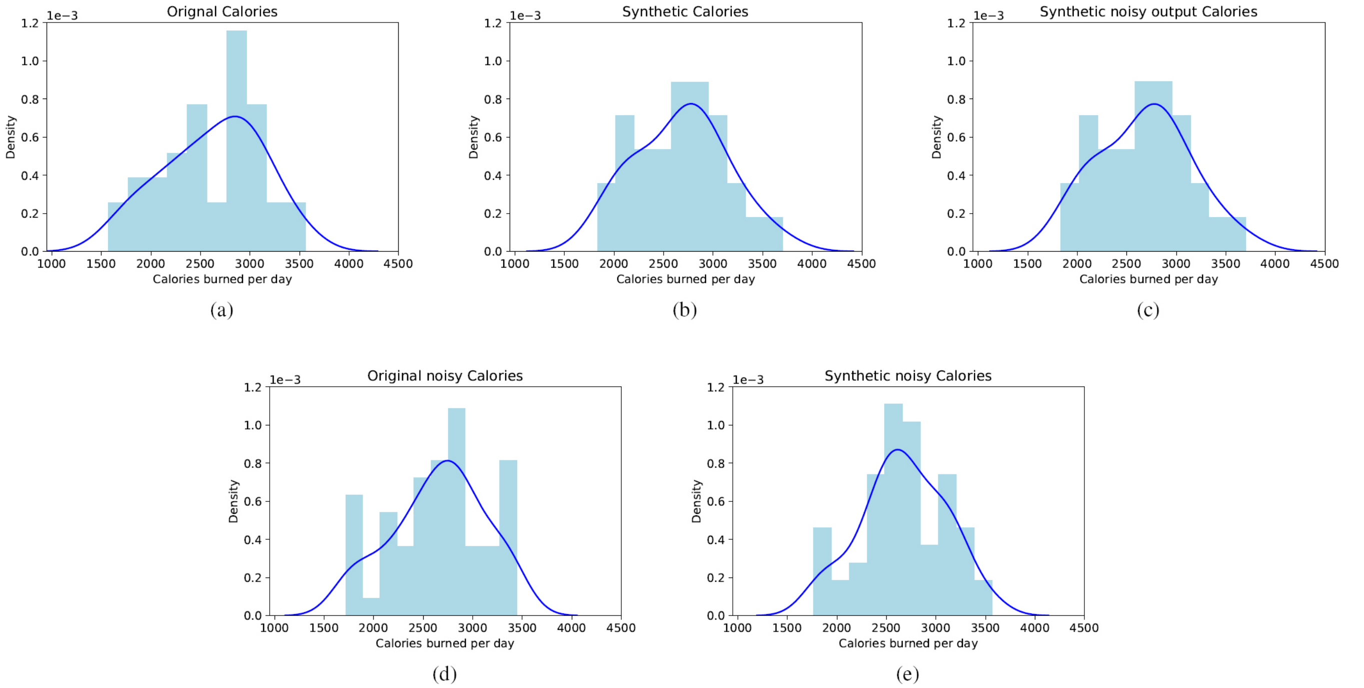
B. Results and Discussion

Our proposed approach generates synthetic and private smart health care data using BGAN in combination with DP. The GAN network is stable, and able to generate plausible results. Figure 5 shows the stability of the proposed GAN where the top subplot shows line plots for the discriminator loss for real samples (blue), the discriminator loss for generated fake samples (orange), and the generator loss for generated fake samples (green). It can be seen that the three losses are somewhat unstable early in the run before stabilizing around epoch 420 to epoch 600. Losses remain stable after that, showing the stable behavior of the GAN, although the variance increases. The discriminator loss for real samples and fake samples is around 0.5, and loss for the generator is slightly higher between 0.5 and 1.0. It is expected the model will generate plausible samples between epochs 420 to 600. The bottom subplot shows a line plot of the discriminator accuracy on real (blue) and fake (orange) samples during training. Similar behavior can be seen as seen in the subplot of loss i.e., the accuracy starts off quite different between the two sample types, then stabilizes between epochs 420 to 600 at around 60% to 70%, and remains stable beyond that, although with increased variance.

Table II shows an example of few rows from the real dataset, and the synthetic rows were generated by the trained GANs. Here, *Original* and *GAN* represent datasets samples with no noise addition (non-DP). *GAN with DP output* shows generated data samples with DP-noise addition (noisy output). Similarly, *Original DP* and *GAN with DP input* represent the original DP-input and the generated synthetic samples

Dataset	Height	Weight	Fat	Fiber	Carbs	Sodium	Protein	Calories Burned	Resting HR	Active Minutes				Steps
										Lightly	Moderately	Very	Sedentary	
Original	169	66.18	39.0	11.0	33.0	1189.0	4.0	2308.08	59.526	121	6	28	731	9706
	169	66.18	39.0	7.0	125.0	1125.0	34.0	2707.35	61.99	166	13	56	732	14070
	169	66.18	33.0	8.0	96.0	1361.0	66.0	2485.35	58.45	99	22	60	774	12008
BGAN	175	87.09	29.20	7.80	73.46	1192.83	41.48	2556.28	63.44	157	63	78	768	12222
	175	87.09	41.32	10.71	49.00	1072.10	33.07	2873.35	64.92	195	49	84	772	14374
	175	87.09	60.82	11.69	98.62	1447.21	31.67	3286.82	64.77	253	56	73	889	14877
BGAN w/ DP output	177	93.62	30.51	13.55	70.19	1191.99	42.48	2555.69	66.29	154	60	71	772	12222
	177	93.62	37.94	9.63	50.15	1071.65	34.7	2875.86	70.71	195	48	81	772	14374
	177	93.62	62.06	9.84	94.18	1447.51	32.17	3286.16	70.32	252	49	73	892	14875
Original DP	161	66.78	39.04	10.02	33.97	1195.83	5.42	2308.34	57.32	121	8	26	732	9704
	161	66.78	38.06	7.31	125.42	1122.46	32.66	2706.42	67.77	164	9	56	731.91	14074
	161	66.78	29.79	2.82	99.02	1360.55	65.02	2485.37	57.75	97	23	62	777	12005
BGAN w/ DP input	181	80.74	33.14	4.12	99.12	1020.8	39.5	3099.14	58.69	213	51	23	757	9769
	181	80.74	22.25	11.02	38.29	1416.57	4.838	2611.83	58.09	137	2	3	754	9944
	181	80.74	58.99	11.19	104.60	483.94	41.96	2593.24	59.56	190	48	106	732	12004

TABLE II: Example data samples from Belgium population. `Age` and `Gender` are hidden.



(a)

(b)

(c)

(d)

(e)

Fig. 6: Histogram distributions for calories burned per day (kcal). Samples: Belgium males with `RHR=70-75bpm`.

respectively (noisy input). As can be seen, all the generated examples look realistic in all the selected privacy settings.

In order to see if the generated data and the real data both come from the same distribution, we show a visualization of the respective histograms. As an example, we only consider the distribution of the burned calories from the logs of male participants belonging to Belgium. It can be seen from Fig. 6 that the original (Fig. 6 (a)) and the synthesized (Fig. 6 (b)) burned calories follow more or less the same kind of distribution indicating the good performance of the proposed BGAN network. We also perform the Kolmogorov–Smirnov (KS) goodness of fit test [31] on the samples taken from original and synthetic calories distributions, which gives a *p-value* of 0.98, indicating a high probability that these samples are from the same distribution and showing that the proposed BGAN is indeed able to learn the diverse categorical and numerical features and generates realistic synthetic samples.

The distribution of original DP data samples (noisy input) is depicted in Fig. 6(d), which exhibits a similar distribution as the DP data generated by BGAN shown in Fig. 6(e). Moreover, the KS test on original noisy calories distribution (DP input) and the synthetically generated noisy calories distribution gives a *p-value* of 0.97, indicating a high probability that the samples come from the same distribution and the proposed BGAN is able to generate differentially private sample distributions.

Using DP input allows us to generate differentially private data instead of explicitly applying DP to all the synthesized data samples. On the other hand, applying DP-mechanism after synthetic data generation allows for more control in terms of noise addition, and consequently, data utility. As can be seen in Fig. 6(c), the distribution of the samples is retained although the records are noised and differentially private.

All our experiments and results demonstrate that although the proposed GAN architecture is quite simple, yet it achieves very high performance in terms of generating both synthetic and differentially private synthetic data. The Fitbit-based smart health care dataset possesses highly diverse features and the proposed DP-mechanism with BGAN is stable and generates high utility synthetic data.

VI. Conclusions

We have proposed a system for creating synthetic and private smart health care datasets using BGANs and differential privacy. Using a real-world collection of Fitbit-based smart health care datasets, we tested our proposed approach in three privacy preservation settings. Our proposed approach is able to learn categorical and numerical values for highly diverse tabular data distributions, and we obtain stable GANs trained for dataset generation. As a result, we generate realistic synthetic smart health care datasets that possess similar distributions as the real data while preserving user privacy. Our proposed method for smart health care data generation also allows control for different privacy settings and paves way for the publication of open smart health care datasets for sharing and use in research and industry.

Acknowledgment

The authors are truly grateful to the anonymous participants who volunteered to provide their data for this study.

References

[1] N. Ahmad, R. P. George, and R. Jahan, "Emerging trends in iot for categorized health care," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, 2019, pp. 1438–1441.

[2] R. Lee, R. Jang, M. Park, G. Jeon, J. Kim, and S. Lee, "Making IoT data ready for smart city applications," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, pp. 605–608.

[3] R. Dagar, S. Som, and S. K. Khatri, "Smart farming – IoT in agriculture," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018, pp. 1052–1056.

[4] N. Scarpato, A. Pieroni, L. Di Nunzio, and F. Fallucchi, "E-health-IoT universe: A review," *management*, vol. 21, no. 44, p. 46, 2017.

[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[6] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, p. 20180083, 2018.

[7] J. Walonoski, M. Kramer, J. Nichols *et al.*, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 230–238, 2018.

[8] H. Li, L. Xiong, L. Zhang, and X. Jiang, "DPSynthesizer: Differentially private data synthesizer for privacy preserving data sharing," *Proc. VLDB Endow.*, vol. 7, no. 13, p. 1677–1680, Aug. 2014. [Online]. Available: https://doi.org/10.14778/2733004.2733059

[9] M. Young, L. Rodriguez, E. Keller, F. Sun, B. Sa, J. Whittington, and B. Howe, "Beyond open vs. closed: Balancing individual privacy and public accountability in data sharing," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 2019, pp. 191–200.

[10] M. K. Baowaly, C.-C. Lin, C.-L. Liu, and K.-T. Chen, "Synthesizing electronic health records using improved generative adversarial networks," *Journal of the American Medical Informatics Association*, vol. 26, no. 3, pp. 228–241, 2019.

[11] H. Ping, J. Stoyanovich, and B. Howe, "Datasynthesizer: Privacy-preserving synthetic datasets," in *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, 2017, pp. 1–5.

[12] S. Imtiaz, R. Sadre, and V. Vlassov, "On the case of privacy in the IoT ecosystem: A survey," in *2019 International Conference on Internet of Things (iThings)*. IEEE, 2019, pp. 1015–1024.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[14] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Advances in Neural Information Processing Systems*, 2019, pp. 7335–7345.

[15] R. D. Hjelm, A. P. Jacob, T. Che, A. Trischler, K. Cho, and Y. Bengio, "Boundary-seeking generative adversarial networks," *arXiv preprint arXiv:1702.08431*, 2017.

[16] A. Torfi and E. A. Fox, "COR-GAN: Correlation-capturing convolutional neural networks for generating synthetic healthcare records," *arXiv preprint arXiv:2001.09346*, 2020.

[17] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama, "GAN-based synthetic brain MR image generation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 734–738.

[18] J. Jordon, J. Yoon, and M. Van Der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2018.

[19] R. Torkzadehmahani, P. Kairouz, and B. Paten, "DP-CGAN: Differentially private synthetic data and label generation," in *Proceedings of the IEEE CVPR Workshops*, 2019, pp. 0–0.

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

[21] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren, "GANobfuscator: Mitigating information leakage under GAN via differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2358–2371, 2019.

[22] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," *arXiv preprint arXiv:1706.02633*, 2017.

[23] Y. Qu, S. Yu, J. Zhang, H. T. T. Binh, L. Gao, and W. Zhou, "GAN-DP: Generative adversarial net driven differentially privacy-preserving big data publishing," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[24] 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 *Machine learning for healthcare conference*. PMLR, 2017, pp. 286–305.

[25] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *FnT-TCS*, vol. 9, no. 3-4, pp. 211–407, 2014.

[26] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[27] C. M. Bowen and J. Snoke, "Comparative study of differentially private synthetic data algorithms from the NIST PSCR differential privacy synthetic data challenge," 2020.

[28] M. Neunhoeffer, Z. S. Wu, and C. Dwork, "Private post-gan boosting," *arXiv preprint arXiv:2007.11934*, 2020.

[29] Q. Geng and P. Viswanath, "The optimal noise-adding mechanism in differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 925–951, 2015.

[30] Nutritionx API. [Online]. Available: https://developer.nutritionix.com/. Accessed 2021-02-23.

[31] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.