Leveraging Synthetic Data for Enhanced Healthcare Analytics

Nuno de Vasconcelos Machado
94021

Supervisors: José Maria Moreira
Jorge Cerejo

***Abstract*—The potential of synthetic data in healthcare analytics, especially as a solution to privacy and data scarcity concerns, is gaining recognition. This project explores and compares various synthetic data generation techniques within this context. We focused on two datasets in the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) format, utilising a single table representation at the visit level. Our exploration included Generative Adversarial Networks (GANs) and fine-tuned Generative Pre-trained Transformers (GPTs), selected for their ability to handle complex data structures. To assess the fidelity of the synthetic data, we employed Log-Cluster and Kullback-Leibler (KL) Divergence between co-occurrence matrices of pairs of concepts metrics. We used prevalence plots for a visual dataset-level comparison. The results with these plots and the Log-Cluster metric were positive and aligned with the literature. However the results regarding the relations between concepts were not as impressive. A noteworthy outcome was observed with a GPT-2 model fine-tuned on our single table format, achieving a KL divergence score of 1.82 and a Log-Cluster score of -6.24. These findings, while divergent from expectations, provide valuable insights into the efficacy of different synthetic data generation methods in healthcare analytics.**

I. Introduction

During the past decade, it has been clear the impact and power of data in technology. Specifically, the sensitive nature of healthcare data poses challenges for the development of tools and models when compared to other fields [1]. Electronic Health Records (EHRs) are high-dimensional, tabular compilations of patient health information that include a wide array of categorical data elements such as demographics, diagnoses, medications and lab results. These challenges on top of all the privacy constraints related to the field make data availability difficult to the broader Machine Learning (ML) community. Synthetic data can help tackle these issues in several ways, either by augmenting existing datasets or minority classes in data to build more robust models, and to replicate them in a relevant statistical way while preserving patients' delicate information [2]. The goal of this project was to find adequate open-source tools in *Python* to generate synthetic data, compare them via relevant metrics and discuss their differences. This project was developed at Hospital da Luz Learning Health as a project for the course Applications of Data Science and Engineering for the Master Degree in Data Science and Engineering at Instituto Superior Técnico, Lisbon.

II. State of The Art

Healthcare data varies significantly in its storage format, such as differences in column names and relationships within a database, depending on the purpose of collection, the organization managing it, and its geographical location. The Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) [3] arose from the need to standardize medical datasets. The major advantage of a standardized format are plug-in tools able to perform relevant data analysis tasks, like assessing data quality and building predictive models. These tools gain added significance in fields like medicine, where the principle of patient privacy is paramount. The ability to utilize these tools locally is crucial, as it ensures the confidentiality and security of sensitive medical information while facilitating effective data analysis and management within the healthcare environment. The core table in the OMOP-CDM representation is the `Person` table which stores each patient basic information. The table `Visit_Occurrence` stores the general information regarding each visit by a patient. There are multiple tables linked to this one describing all the types of events that can happen within a visit among them there are the tables `Procedure_Occurrence`, `Condition_Occurrence`, `Measurement` and `Drug_Exposure` [3]. Standardized vocabularies and terminologies are also a crucial aspect of this representation [3]. The two datasets used in this project were in this format already. Figure 1 presents a diagram of part of its schema.

Synthetic data generation tools gain added significance in fields like medicine, where the principle of patient privacy is paramount. The ability to utilize these tools locally is crucial, as it ensures the confidentiality and security of sensitive medical information while facilitating effective data analysis and management within the healthcare environment.

Synthetic data can be defined as microdata records generated by a model that is trained to reproduce the real data statistical properties [4]. The aim is for this synthetic data to approximate these properties without precisely copying the original dataset. Creating effective synthetic data, however, presents challenges, such as ensuring data fidelity, where the synthetic version accurately represents the complex patterns and variability of the real data, and maintaining privacy, where the data does not inadvertently reveal sensitive information.

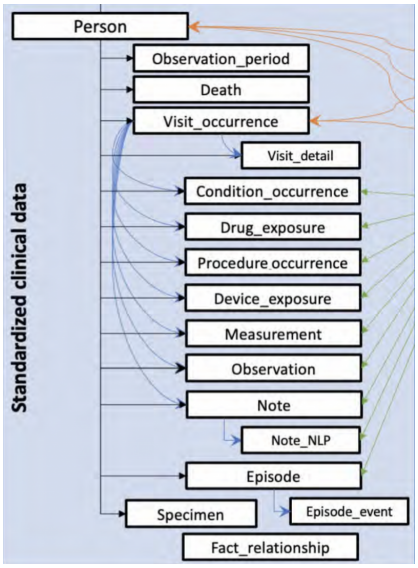

Fig. 1: Part of the OMOP-CDM database schema [3]

Synthetic data can be used as a way to accelerate the availability of data resembling the real one, so that models and other tools can be developed sooner with less data protection concerns [4]. Furthermore, these models are also used to augment existing datasets, contributing to the creation of more robust models.

There are several types of models capable of generating synthetic data, ranging from more classic approaches based on mixtures of distributions to more recent Generative Adversarial Networks (GANs) and transformers based architectures [5].

More used in a financial data context, Gaussian Copulas focus solely on the dependency structure between variables, irrespective of their individual distributions [6]. This is why they are particularly versatile for capturing relationships in datasets where variables have diverse distributions. Although the reliance on a normal distribution framework can be a limitation in scenarios where the relationships between variables are not linear or exhibit non-normal characteristics [7].

GAN architectures have achieved incredible results in the generation of synthetic data, particularly in the domains of images and text data [8]. In GANs, two models are competing against each other, one trying to generate realistic synthetic data (G) and another one trying to discriminate (D) between real and synthetic data. There are adaptations to deal with categorical data like Conditional Tabular GANs (CT-GANs) and Anyway CTGANs (ACTGANs). In Gonçaslves et al. [5], the variant of GANs used presented disappointing results, suggesting that its impact in the context of eletronic Health records (EHR) data may still be limited.

Recent advancements in synthetic data generation models have focused on direct integration with data in the OMOP-CDM format. These models leverage transformed base architectures, converting the data into a patient-level single table representation using it in predictive tasks [9]. In this representation, the patient's demographic information, as well as all

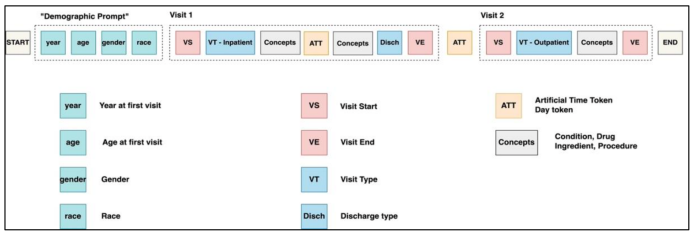

Fig. 2: Patient level representation [10]

the events associated with each episode are kept (conditions, procedures and drugs administrated). The time information is tracked by using Artificial Time Tokens (ATT's) that are added between visits [9]. As a way to lose less time information the patient level representation in [10] also adds ATT's between the occurrences inside each visit. Figure 2 shows a diagram of this representation in a patient with two visits. The ATT's used correspond to discretisations of the time intervals between events so there is still time information being loss.

In Pang et al. [10], a concept and a contextual embedding are applied to this representation before 6 standard transformer decoders are trained with 3 million unique patients' medical histories. The synthetic data is then sampled by prompting the trained model with demographic information from real patients. Several sampling strategies were used. Top p-value where the sampling of each token is done from the most probable tokens with a cumulative $p\%$ probability. Top k-value where the sampling is done from the top k most probable tokens [11]. Both of these were compared. In reference [12], a Variational Auto Encoders (VAE) with a GAN architecture uses this same patient level representation. However, in this case the embedding is trained within the VAE model. Both of these seem promising advancements in the field however the code for both is not available at the time of writing.

Generative Pre-trained Transformer(GPT) models, trained on extensive datasets, are highly adept at grasping and mimicking complex data patterns. They can be fine-tuned for particular tasks like creating synthetic data. Transformers have significantly transformed the way sequential data is processed. Unlike earlier methods, Transformers excel at recognizing long-range dependencies within data sequences. This enhanced capability is largely due to their attention-based mechanism, which also allows them to operate in parallel, enhancing processing efficiency [13].

Even though there is no consensual metric to evaluate synthetic data in healthcare, we can combine the several different ones to get an overall idea of the quality of the generated data. In Goçalves et al. [5], the proposed metrics evaluate the data utility at an attribute and at a dataset level, as well as regarding the information disclosure. However, metrics regarding patient's privacy based on similarity are put into question in recent literature [14].

There are several open source *Python* libraries that specifically deal with modelling Synthetic Data, however most of these only deal with data in a single table format wile the

OMOP-CDM is a relational formaat. The library *Synthetic Data Vault* [15] is highlighted as they make most of their tools available to run locally, this library was used in this project. Other libraries such as *Synthea* [16], YData-Synthetic [17] and *Gretel* [18] were also explored, only the ladder also being used in this project to test an approach working directly with a relational dataset.

III. Methods

A. Datasets

1) MIMIC: The MIMIC-IV demo is subset of 100 patients of the MIMIC-IV dataset in the OMOP-CDM format was used. The MIMIC (Medical Information Mart for Intensive Care) dataset is a freely available database comprising de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center in Boston, Massachusetts, USA. It includes information such as demographics, vital signs, laboratory tests, medications, and more [19].

2) Learning Health (LH): A dataset of 1000 patients at *Hospital da Luz Lisboa*, the patients were randomly chosen from a set of patients with at least one visit after 2019 in order to have richer medical data, considering that in 2019 the hospital started to use standard codes (namely, ICD9, Loinc) more frequently, allowing an easier mapping to the OMOP-CDM.

These datasets defer in size but also in nature. The *MIMIC* corresponds to mostly data from the Intensive Care Unit which is much richer in the number of events per visit. While the *LH* dataset has more visits per patient, however these have less events, given that they may be any type of a patient episode in the hospital. This way, the LH dataset captures better the medical history of a patient.

B. Synthetic Data Generation directly from the OMOP-CDM format

This first approach focused on finding tools that dealt with a relational dataset comprised of several tables like the OMOP-CDM representation being used. This way, we would be able to directly capture the relations between concepts, like diseases and their medication. This proved to be a difficult task since most of the these tools were not open source, and the few that were, were not capable of handling very complex data. A model named *Amplify* from the Gretel library was explored in the *MIMIC* dataset. *Amplify* is described in *Gretel*'s documentation as a statistical based simple model with low accuracy but being computational efficiency [20], so it can bring value as a baseline. These models only work with connectors to databases. Therefore a local database with the `.csv` from the MIMIC dataset was set up.

C. Converting the data to a single table representation

The models that directly deal with relational data simply use the links between them to create a single table view from where the synthetic data is then generated. As a way to have more control over how this is done and to be able to use more open source models that only deal with singular tables, we decided to create a single table representation of the data similar to the one in [9]. However, adaptations were necessary due to specific characteristics of the *MIMIC* dataset. In this dataset, most patients had a limited number of hospital visits, with the majority having only one. Additionally, the total number of patients in our datasets was significantly smaller, consisting of 100 and 1000 patients, respectively, in contrast to the 3 million patients in the dataset referenced in [9]. Consequently, employing a similarly complex representation as in the larger dataset did not yield satisfactory results, as was confirmed through preliminary testing. In this representation the data is presented at patient's visit level, where every row is a different visit consisting of all the codes associated with it from the `Drug_Exposure`, `Condition_Occurrence` and `Procedure_Occurrence` tables, the year of birth and gender from the `Person` table, the patient's age at the start of the visit as well as the visit type. Figure 3 presents a diagram of this representation. Compared to the original approach we are losing time information between visits in a patient medical history, however it's a simpler representation that might be fitting especially when dealing with a smaller dataset.

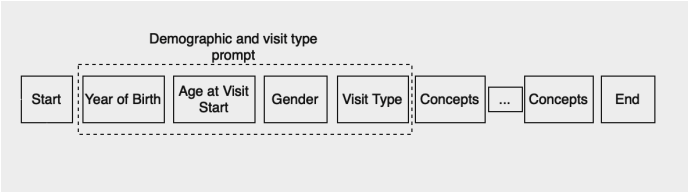

Fig. 3: Visit level representation used

D. Finetuning a GPT-2 model

As mentioned in Pang et al. [10], a GPT model was trained from scratch with 6 transformer decoder units with their patient level representation being used after applying a concept and a positional embedding. Ideally, to replicate this approach with our visit level representation we would also train a generative model from scratch. However, considering that we have a lot fewer patients and limited computational resources, we tried a simpler approach where a pre-trained GPT-2 model was fine tuned on our data. Even though the GPT-2 model was trained on textual data it captures general information regarding patterns, relations and contextual information that can be useful in our context. Although it is not the ideal approach it can be interesting as a way to replicate the paper in a less computational demanding way, while also testing to some extent our simpler visit level representation.

After the model was trained, synthetic data was generated by prompting the model with the demographic and visit type data in two ways. One by randomly selecting patient's data from the real data and other by randomly sampling each individual attribute from the real patients data. The results of the two methods on the MIMIC dataset were similar so on the LH dataset we only used the individual attribute sampling approach. In both approaches, we implemented a

top-p sampling strategy, setting the threshold at $p = 80\%$. This involved selecting each new token from a pool of the most probable tokens, which collectively account for a cumulative probability of 80%. Each generated data sample was cut off at the first occurrence of the end token. The model was generating concepts not present in the real data. So even if these corresponded to real concepts it would be a mere coincidence so they were removed from the analysis. The number of made up concepts generated in the whole dataset was used as an hallucination metric. When applying this model to the LH dataset it was also noticed that a lot of concepts present in the real data were not being generated at all in the synthetic one. As a way to reduce these cases and also make the computations more efficient all the concepts that appeared less than 10 times were removed, as well as the duplicates in each individual visit. This was only done in the LH dataset.

E. Synthetic Data Generation directly from our visit level representation

To enable the use of our variable-length visit level representation in non-transformer based models, we employed one-hot encoding. This encoding was done without including the tokens that are specific to GPT models. In this encoding scheme, each column represents a different concept code, with the presence or absence of a concept in a visit indicated by 1 or 0, respectively. We are of course losing information regarding visits where a concept repeats itself, however if the goal is to only model the unique concepts of a visit this is fitting. In the *LH* dataset the concepts that appeared less than 10 times in the whole dataset were removed to make the modeling more computationally efficient. In a sort of way we are removing the outliers from the data, which can be considered to be the patients with the more sensitive data, so this can be also thought of as a privacy preserving measure. The models used were Gaussian Copulas and CTGANs, both from the *SDV* library [15] [21]. Both models were applied to each of our datasets.

F. Metrics

Prevalence Plots will be used to compare the frequency of a particular diagnosis, medication, procedure, or visit type between the real dataset and the synthetic one generated by a given method. Both axis are on a logarithmic scale for an easier visual comparison. Points that fall along the dashed diagonal line indicate a perfect match between the source and synthetic data. These plots allow to assess if the synthetic data maintains the statistical characteristics of the original data at a **dataset level**. The dashed line in these plots shows where the points would lie if the synthetic data had exactly the same prevalence as the real data.

The **Log-Cluster** metric captures the similarity between the latent structures of real and synthetic data [5]. This is done by firstly merging the two datasets, while keeping track of each data point's origin. Then, applying the *k-means* clustering algorithm and lastly assessing the ratio between real and synthetic data points in each cluster. Ideally, this ratio would be similar for all clusters and to the overall ratio of synthetic to real data points, meaning that both latent structures are indistinguishable. The metric is defined by equation 1. $G = 4$ is the fixed number of clusters we chose. This value is not very relevant since the clustering itself is not important only the constitutions of each. n_j and n_j^R are respectively the total number of data points and the total number of real data points in the *j*-th cluster. $c = \frac{n^R}{n^R+n^S}$ is the ratio in the full dataset. The lower the value the better.

$$U_c(X_R, X_S) = log(\frac{1}{G}\sum_{j=1}^{G}[\frac{n_j^R}{n_j} - c]^2) \tag{1}$$

The **Kullback-Leibler (KL) Divergence between co-occurrence matrices of pairs of concepts** score. Firstly, for each dataset the co-occurrence matrix between pairs of concepts was obtained and normalized. Then, the KL divergence between the two was calculated which gives a measure of the similarity between the two. The best possible value is 0, which means that the two matrices were equal. This metric gives a score for how well the relationships between pairs of concepts were captured.

As a way to assess the number of concepts generated for each visit occurrence in relation to the real data some statistics regarding the number of concepts per visit were obtained like the average and median. These are more relevant to the GPT-2 model approach to check if the tendencies for this number are captured in the synthetic data.

IV. Results and Discussion

In this section only Prevalence Plots for the concepts in the *Drugs* domain are displayed. The plots for the *Condition*, *Procedure* and *Visit Type* domains were also obtained but considering that their analysis is similar they are only displayed in appendix A. Table I presents a broad comparison between the different methods and datasets via the Log-Cluster and the KL divergence between co-occurrence matrices of pairs of concepts. Some general statistics between the number of concepts in a visit are also presented.

A. MIMIC dataset in relational form

Figure 4 (A.4, B.4) shows that the *Amplify* model directly applied to the relational dataset was able to replicate the prevalence in the original data in a meaningful way. The points are more scattered in the lower prevalence regions (10^{-4}) and tend to fall on the dashed line for the higher frequency region. The straight line the data points appear to form on the lower and left bounds are due to the small size of the dataset and are given by points that appear only once in either one of them. In this approach all the concepts that appear in the real data also appear in the synthetic data. So some level of replication might be happening.There are more data points that fall on the upper part of the dashed line, meaning that there was some over-representation of concepts in the synthetic data. The reasons for this are unclear, especially because no description for the actual model being used is provided in the documentation [20].

TABLE I: Metrics comparison between the different models in the different datasets

Dataset	Representation	Model	Log-Cluster (↓)	KL divergence (co-occ)($\downarrow_0$)	No. of concepts per visit			
					Min	Median	Max	Avg
MIMIC	Relational	Gretel's Amplify	**-3.87**	2.23	-	-	-	-
MIMIC	Visit level	SDV CT-GAN	-1.42	2.78	28	53	110	53.59
MIMIC	Visit level	SDV Copula	-2.70	**2.13**	22	56	124	57.19
MIMIC	Visit Level	GPT-2 - Attribute sampling	-1.57	2.57	0	21	44	20.42
MIMIC	Visit Level	GPT-2 - Person row sampling	-1.57	2.57	0	21	44	21.57
MIMIC	Visit Level	REAL DATA	-	-	1	49.5	188	55.33
LH	Visit Level	SDV CT-GAN	-2.63	6.90	0	2	14	2.64
LH	Visit Level	SDV Copula	**-7.56**	7.20	0	2	9	1.99
LH	Visit Level	GPT-2	-6.24	**1.82**	1	1	13	2.00
LH	Visit Level	REAL DATA	-	-	1	1	28	1.96

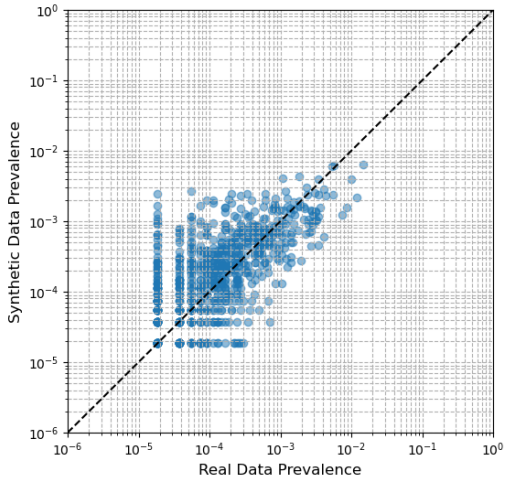

Fig. 4: *Amplify* model in the MIMIC dataset (relational) - Prevalence plot of the domain *Drug*

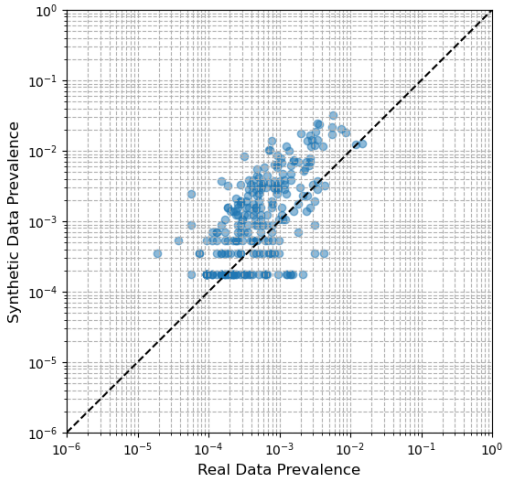

Fig. 5: GP2 model with attribute sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Drug*

B. MIMIC dataset in visit level representation

Figures 5 (A.5, B.5) and 6 (A.6, B.6) prevalence plots show that the results for both sampling techniques with the fine tuned GPT-2 model are similar. They both under-represent the less frequent concepts while over representing the more frequent ones. This means that the model was not very good at capturing the frequency of the rarer concepts in the data. This makes sense considering it is a big model fine tuned with only $\approx$ 200 visit examples. The post processing done also plays a part here. All the concepts that did not appear in the real data set were removed. This skewed the more frequent data points that were probably represented in an accurate way beforehand. The generated data by both sampling methods is very similar, meaning that the sampling strategy had little effect in the concept generation in the dataset as a whole.

The prevalence plot obtained from the CT-GAN model (Figure 7 (A.7, B.7)) displays a notable symmetry with the dashed reference line. This plot, like the others, more accurately represents frequently occurring concepts while exhibiting greater variability in less common ones. In contrast, the plot generated using the Gaussian Copulas method (Figure 8 (A.8, B.8)) appears more constricted, particularly in the higher frequency area. This suggests that the Gaussian Copulas method tends to replicate data more closely than other methods regarding the concepts' frequency.

C. LH dataset in the visit level representation

The lower number of data points on the left hand side of the prevalence plots (Figures 10 (A.10, B.10) and 9 (A.9, B.9)) in the *LH* dataset can be explained by the pre-processing that was done, where all the concepts that appear less than 10 times in the dataset were removed. Figure 9 (A.9, B.9) plot for the fine-tuned GPT-2 model manifests the same behaviour as the prevalence plot for this model in the MIMIC dataset.

The prevalence plot for the CT-GAN model is more chaotic. Both the less and the more common concepts matched in

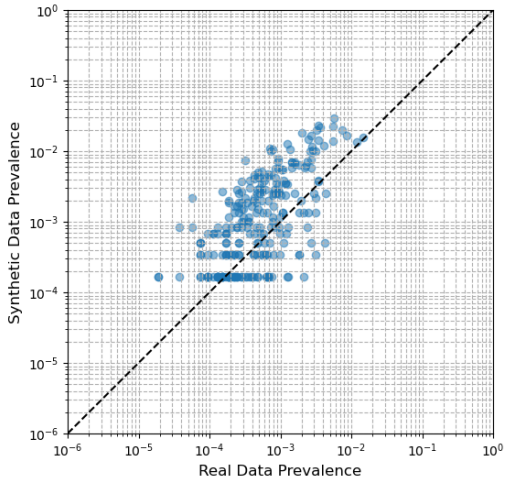

Fig. 6: GP2 model with full `Person` row sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Drug*

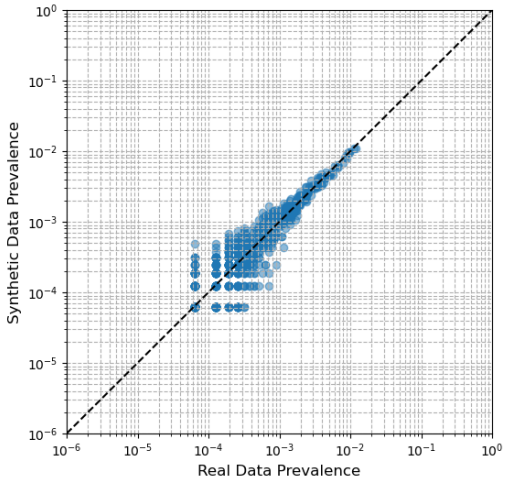

Fig. 8: Gaussian Copulas in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Drug*

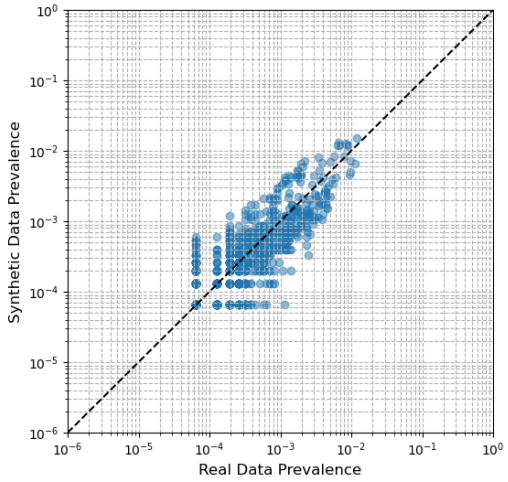

Fig. 7: CTGAN in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Drug*

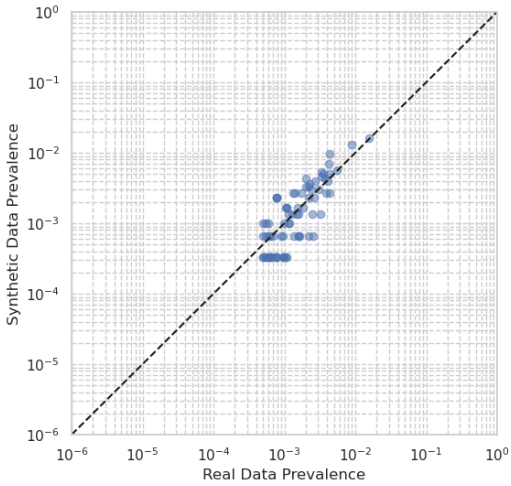

Fig. 9: GP2 model with with attribute sampling in the LH dataset (visit level representation) - Prevalence plot of the domain *Drug*

terms of frequency, while the ones in the middle are more scattered. The reasons of this erratic behaviour are difficult to pinpoint. Possible causes are limitations within the model itself, considering that in the MIMIC dataset this model also presented a broader range of values. The one hot encoding applied to the data to use this model might also play a role here, since there were no repetitions allowed for concepts inside a visit. However the Gaussian Copulas prevalence plots do not follow this behaviour. The 10 times concepts occurrence cut-off might also be at fault.

The prevalence plot for the Gaussian Copula model (Figure 11 (A.11, B.11)) presents a similar behavior to the data in the MIMIC dataset, meaning that it probably is replicating the original data very closely, which is not ideal to maintain the privacy of the real data.

The *age at visit start* distribution between the real data and the data generated by the Gaussian Copulas are very similar (Figure 14). The same for the finetuned GPT-2 model (Figure 12). The number of data points generated with this model was only about 10% of the original dataset size since the generation step was much more computationally demanding in this model. For the CT-GAN (Figure 13) the distributions are also similar, except for the first bin, meaning that the model over represents the patients with ages between 0 and 5 years old, reasons for

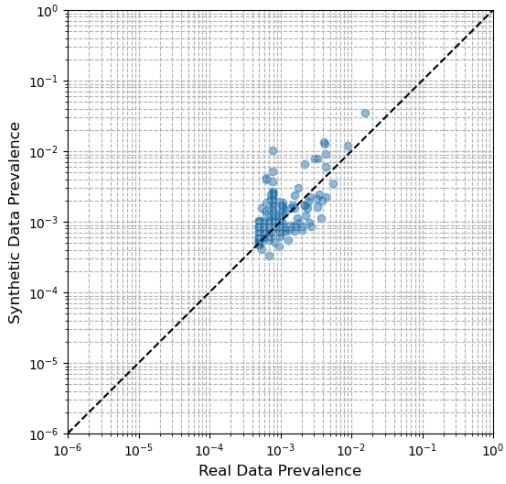

Fig. 10: CTGAN in the LH dataset (visit level representation) - Prevalence plot of the domain *Drug*

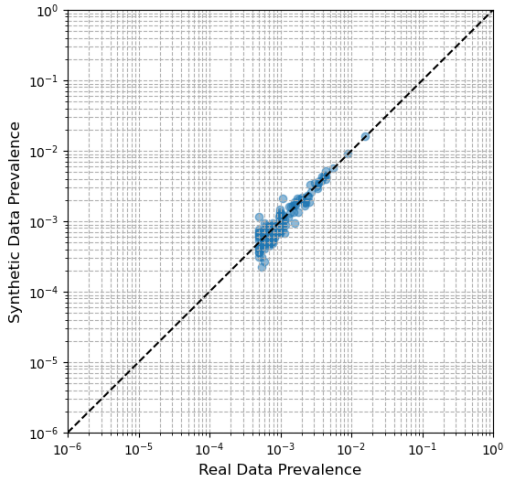

Fig. 11: Gaussian Copulas in the LH dataset (visit level representation) - Prevalence plot of the domain *Drug*

this are unclear but might rely on the model inadequacy with this type of data

Regarding the methods applied to the MIMIC dataset the values for both the Log Cluster metric and the KL divergence score were all similar. The best result in the Log-Cluster metric was obtained for the *Gretel*'s *Amplify* model directly applied to the relational data. This model has access to all the data while the others lose some information regarding visits' time intervals after the conversion to the visit level representation. This might be the reason for the better value, even though the difference is very small. Gaussian Copulas presented the best value in the KL divergence score which can be due to them being able to better capture the relations between pairs

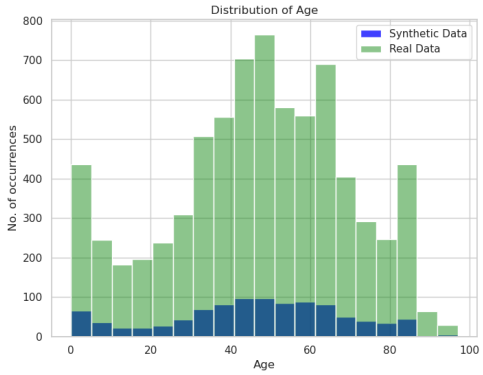

Fig. 12: GP2 in the LH dataset (visit level representation) - Patient's age at visit start distributions comparison

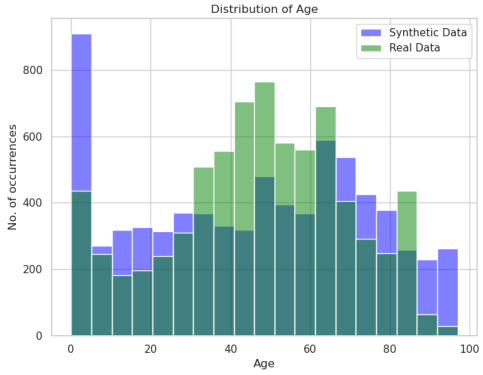

Fig. 13: CTGAN in the LH dataset (visit level representation) - Patient's age at visit start distributions comparison

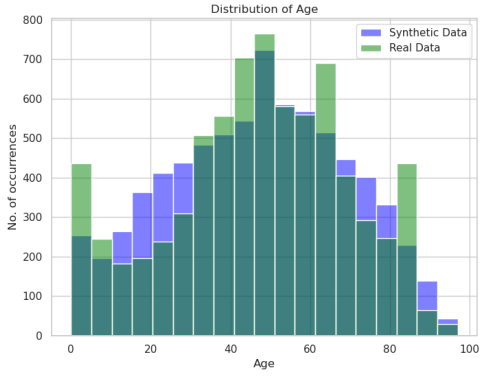

Fig. 14: Gaussian Copulas in the LH dataset (visit level representation) - Patient's age at visit start distributions comparison

of concepts, which they are typically good at doing for linear relations or due to the model replicating the original data more closely. Once again, these differences in values are very small meaning that there is no clear cut better model from the metrics analysis in the MIMIC dataset. The LH dataset results suggest that the fine-tuned GPT-2 model performed considerably better than the other two regarding the KL divergence value, and there was also improvements compared to the same model on the MIMIC dataset. This indicates that the fine tuning with more data brought a better understanding of the model and that even though this was not an ideal approach in terms of using a pre-trained textual model in our concept id's based data the produced results were better than the other models. Reasons for this also lie on the advantages of a transformer based model in terms of the input flexibility. In the other models a one hot encoding representation had to be used, which with the type of datasets at hand inevitably lead to a very sparse representation, which models do not typically deal well with. While a transormer based architecture can handle variable length inputs. The Log Cluster values are also all better than the ones in the mimic dataset, meaning that the models with more data were able to better capture the latent structure of the data. The increases in the KL divergence values (except for the GP2 model) can be attributed to having less events inside each visit in the LH dataset. This means that there is less information regarding the relations between pairs of concepts to be captured so due to the inherent differences between the datasets which make a comparison between the metrics values between the two not fair.

When comparing to the literature the values for the Log-Cluster metric are in the same order of magnitude as the ones in reference [5], where different synthetic data generation tools were also comapred. Although on a simple EHR dataset not in the OMOP-CDM format. So in regards to this metric the results are positive. The KL divergence scores obtained are much worse than the results from reference [10], where values between 0.30 and 0.60 were obtained for this metric. The best value we obtained was 1.82 for the fine-tuned GPT-2 model. Our value being much larger is of course not great, but we have to take in consideration that we used a much simpler representation, a pre-trained model made for textual data and a much smaller dataset. With these factors in mind the results do not seem that negative.

V. Conclusions

Overall, we tried different models for synthetic data generation in the OMOP-CDM format. We also compared them via metrics regarding data fidelity as proposed. The best results were obtained for the fine-tuned GPT2 model. This is in line with the most recent literature, where transformer based models are presented as a promising way to generate synthetic data. Our results with the fine-tuned GPT-2 model are particularly interesting if we take into account that we used mostly codes in our representation and a GPT2 model is pre-trained on textual data. Adjustments to this pipeline could bring even better results. Especially considering it is much less computationally demanding than training a GPT model from scratch.

In the future fine tuning a pre-trained GPT model with the concept descriptions instead of the codes could be interesting. Or even a generated coherent text using the concepts descriptions in a way that was reversible back to the codes format. This could be a way to better leverage the textual capabilities of a pre-trained model. A LLM trained from scratch to deal with medical codes and our visit level representation should yield better results as well. Converting the generated data in the visit level representation back to the OMOP-CDM format is possible, even though with some time information loss. Exploring metrics related to the similarity between the real and synthetic dataset could bring value, even though these are not consensual.

References

[1] A. Tucker, Z. Wang, Y. Rotalinti, and P. Myles, "Generating high-fidelity synthetic patient data for assessing machine learning healthcare software," *npj Digital Medicine*, vol. 3, 2020.

[2] R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. Williamson, and F. Mahmood, "Synthetic data in machine learning for medicine and healthcare," pp. 493–497, 6 2021.

[3] OHDSI, *The Book of OHDSI: Observational Health Data Sciences and Informatics*. OHDSI, 2019. [Online]. Available: https://books.google.pt/books?id=JxpnzQEACAAJ

[4] A. Gonzales, G. Guruswamy, and S. R. Smith, "Synthetic data in health care: A narrative review," *PLOS Digital Health*, vol. 2, p. e0000082, 1 2023.

[5] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, "Generation and evaluation of synthetic patient data," *BMC Medical Research Methodology*, vol. 20, 5 2020.

[6] D. Tjøstheim, H. Otneim, and B. Støve, "Chapter 5 - local gaussian correlation and the copula," in *Statistical Modeling Using Local Gaussian Approximation*, D. Tjøstheim, H. Otneim, and B. Støve, Eds. Academic Press, 2022, pp. 135–159. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128158616000122

[7] P. R. Dewick and S. Liu, "Copula modelling to analyse financial data," *Journal of Risk and Financial Management*, vol. 15, 3 2022.

[8] G. Iglesias, E. Talavera, and A. Díaz-Álvarez, "A survey on gans for computer vision: Recent research, analysis and taxonomy," *Computer Science Review*, vol. 48, p. 100553, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013723000205

[9] C. Pang, X. Jiang, K. S. Kalluri, M. Spotnitz, R. Chen, A. Perotte, and K. Natarajan, "Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks," 11 2021. [Online]. Available: http://arxiv.org/abs/2111.08585

[10] C. Pang, X. Jiang, N. P. Pavinkurve, K. S. Kalluri, E. L. Minto, J. Patterson, and K. Natarajan, "Generating synthetic electronic health records in omop using gpt," 2023. [Online]. Available: https://www.ohdsi.org/2023showcase-503/

[11] I. Vibudh, "A guide to controlling llm model output: Exploring top-k, top-p, and temperature parameters," https://ivibudh.medium.com/a-guide-to-controlling-llm-model-output-exploring-top-k-top-p-and-temperature-parameters-ed6a31313910, 2023, accessed: 2024-01-21.

[12] S. C.-G. A. Labarga, S. Aguiló-Castillo, "Generating synthetic data from omop-cdm databases for health applications," 2021. [Online]. Available: https://f1000research.com/posters/12-695

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 6 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[14] G. Ganev and E. D. Cristofaro, "On the inadequacy of similarity-based privacy metrics: Reconstruction attacks against "truly anonymous synthetic data"," 12 2023. [Online]. Available: http://arxiv.org/abs/2312.05114

[15] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2016, pp. 399–410.

[16] "Synthea: Synthetic patient generation," https://github.com/synthetichealth/synthea, accessed: 2024-01-21.
[17] "Ydata synthetics: Synthetic data generation library," https://github.com/ydataai/ydata-synthetics, accessed: 2024-01-21.
[18] "Gretel: Privacy-preserving synthetic data generation," https://gretel.ai/, accessed: 2024-01-21.
[19] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, B. Moody, B. Gow, L. wei H. Lehman, L. A. Celi, and R. G. Mark, "Mimic-iv, a freely accessible electronic health record dataset," *Scientific Data*, vol. 10, 12 2023.
[20] Gretel.ai, "Gretel amplify," https://docs.gretel.ai/create-synthetic-data/models/synthetics/gretel-amplify, 2024, accessed on January 18, 2024.
[21] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019.

APPENDIX

A. First Appendix

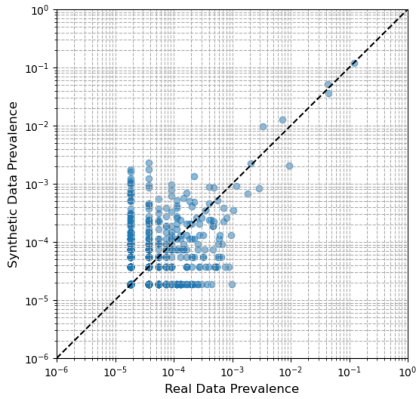

Fig. A.4: Amplify model in the MIMIC dataset (relational) - Prevalence plot of the domain *Condition*

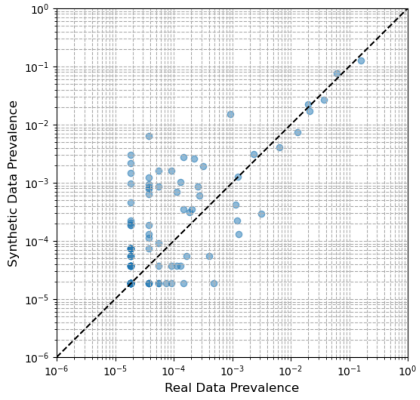

Fig. B.4: Amplify model in the MIMIC dataset (relational) - Prevalence plot of the domain *Procedure*

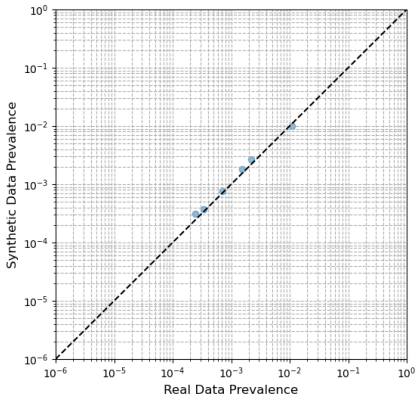

Fig. C.4: Amplify model in the MIMIC dataset (relational) - Prevalence plot of the domain *Visit Type*

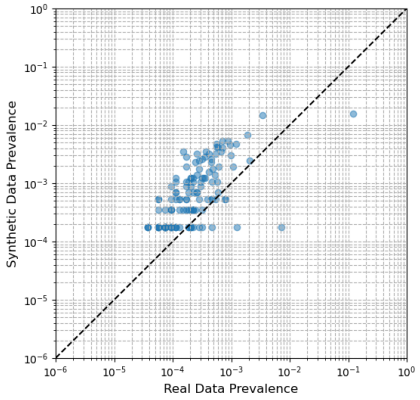

Fig. A.5: GP2 model with attribute sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Condition*

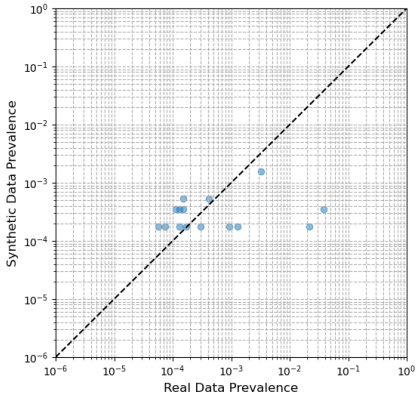

Fig. B.5: GP2 model with attribute sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Procedure*

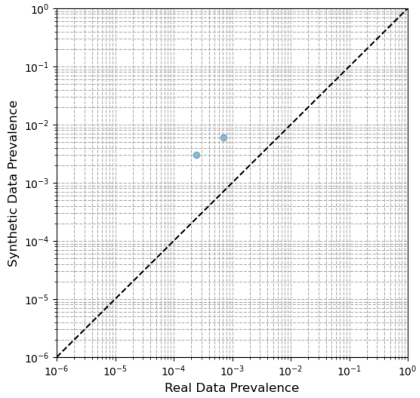

Fig. C.5: GP2 model with attribute sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

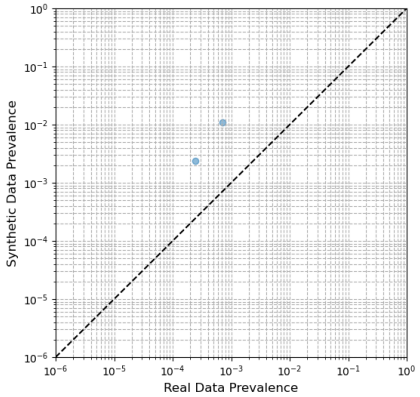

Fig. C.6: GP2 model with full `Person` row sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

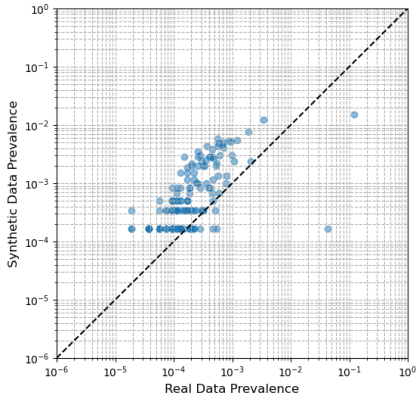

Fig. A.6: GP2 model with full `Person` row sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Condition*

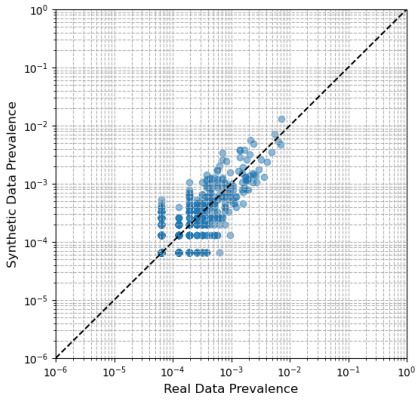

Fig. A.7: CTGAN in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Condition*

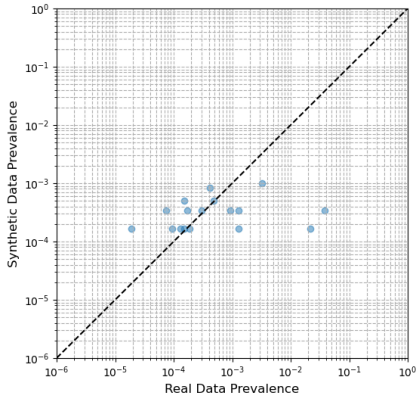

Fig. B.6: GP2 model with full `Person` row sampling in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Procedure*

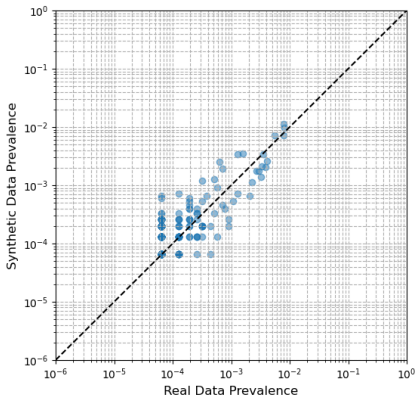

Fig. B.7: CTGAN in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Procedure*

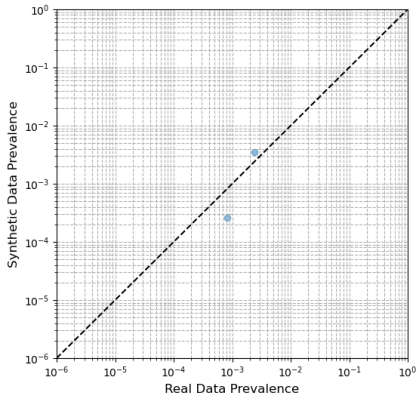

Fig. C.7: CTGAN in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

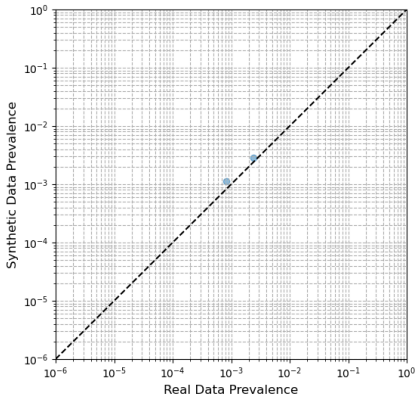

Fig. C.8: Gaussian Copulas in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

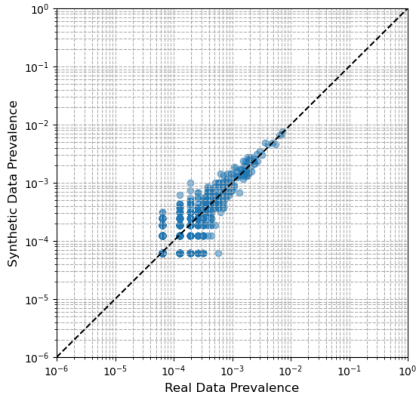

Fig. A.8: Gaussian Copulas in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Condition*

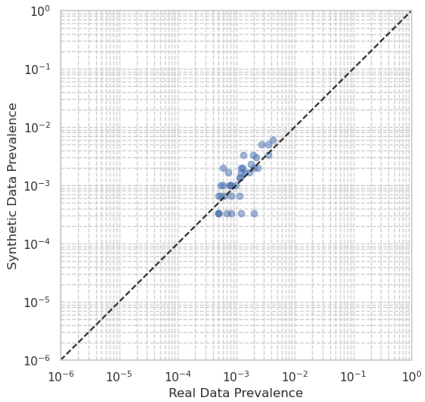

Fig. A.9: GP2 model with with attribute sampling in the LH dataset (visit level representation) - Prevalence plot of the domain *Condition*

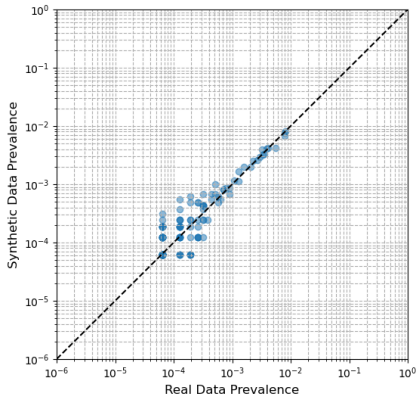

Fig. B.8: Gaussian Copulas in the MIMIC dataset (visit level representation) - Prevalence plot of the domain *Procedure*

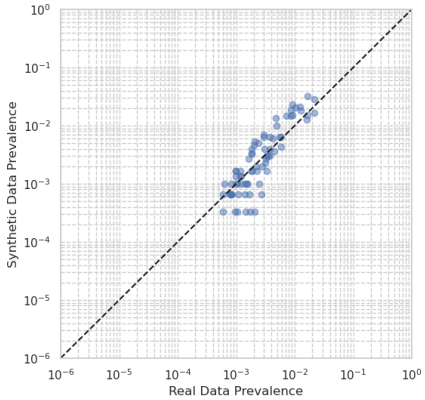

Fig. B.9: GP2 model with with attribute sampling in the LH dataset (visit level representation) - Prevalence plot of the domain *Procedure*

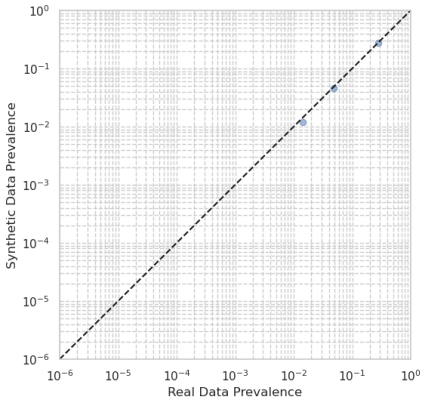

Fig. C.9: GP2 model with with attribute sampling in the LH dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

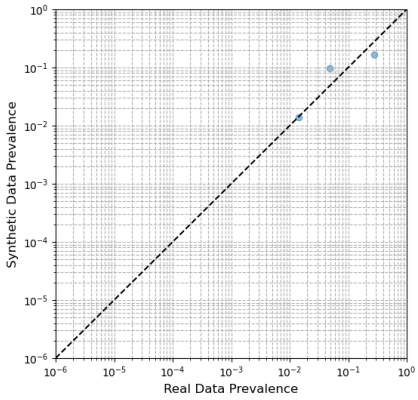

Fig. C.10: CTGAN in the LH dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

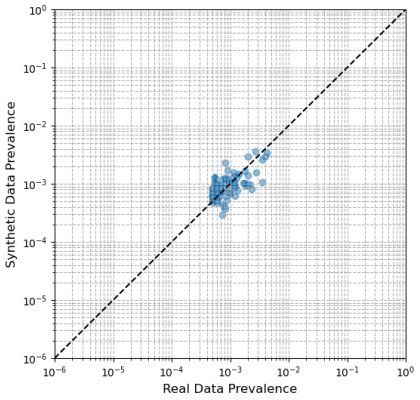

Fig. A.10: CTGAN in the LH dataset (visit level representation) - Prevalence plot of the domain *Condition*

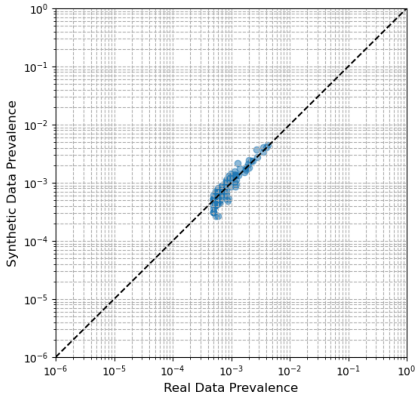

Fig. A.11: Gaussian Copulas in the LH dataset (visit level representation) - Prevalence plot of the domain *Condition*

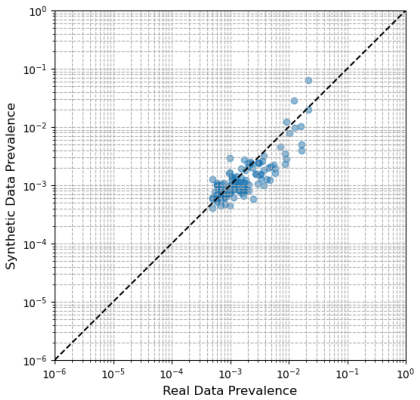

Fig. B.10: CTGAN in the LH dataset (visit level representation) - Prevalence plot of the domain *Procedure*

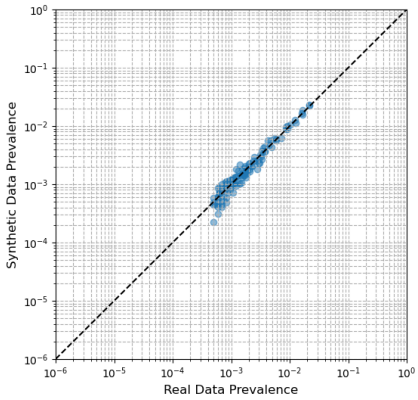

Fig. B.11: Gaussian Copulas in the LH dataset (visit level representation) - Prevalence plot of the domain *Procedure*

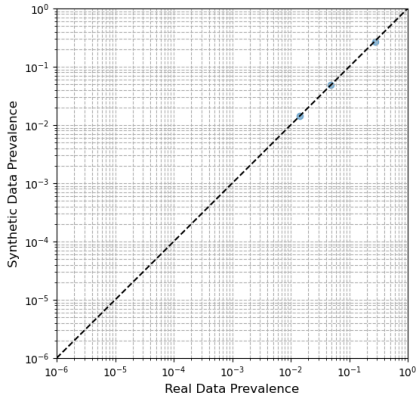

Fig. C.11: Gaussian Copulas in the LH dataset (visit level representation) - Prevalence plot of the domain *Visit Type*

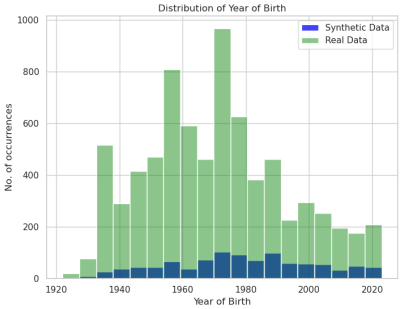

Fig. D.12: GP2 in the LH dataset (visit level representation) - Patient's year of birth at visit start distributions comparison

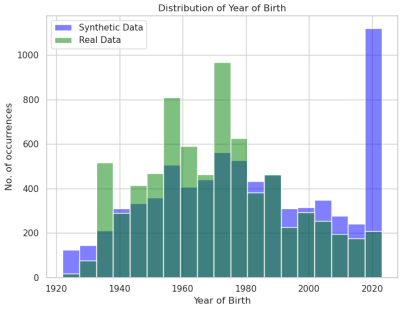

Fig. D.13: CTGAN in the LH dataset (visit level representation) - Patient's year of birth at visit start distributions comparison

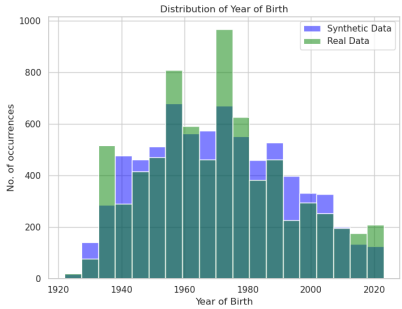

Fig. D.14: Gaussian Copulas in the LH dataset (visit level representation) - Patient's year of birth at visit start distributions comparison