Creating Customers That Never Existed: Synthesis of E-commerce Data Using CTGAN

Melle Mendikowski[1] and Mattis Hartwig[1,2]

[1] German Research Center for Artificial Intelligence, Lübeck
[2] singularIT GmbH, Leipzig

Abstract. Various e-commerce use cases that companies implement in applications rely on personal data of customers. Privacy and data protection play an important role when discussing the usage of personal customer data resulting in a conflicting demand between data collection and data protection. Researchers have found a promising solution to this problem: the generation of synthetic data which is not connected to real people. In this paper, we use the deep learning architecture Conditional Tabular Generative Adversial Network (CTGAN) to synthesize e-commerce data. Especially the categorical relationships between columns of e-commerce data include fixed dependencies, where e.g. an entry in the sub-category column is defining the entry in the category column as well. These specific characteristics result in the necessity to evaluate the suitability of the CTGAN architecture for synthesizing e-commerce data which is the focus of this paper. We present a new similarity measure for synthetic and original datasets that focuses on categorical correlations: the Cramer's V deviation (*CV-deviation*). In our experiments, we create synthetic e-commerce data from a publicly available dataset using CTGAN. We use an existing and our newly developed *CV-deviation* measure in hyperparameter selection and compare the outcomes. By incorporating *CV-deviation* into the performance metric, we manage to increase the ability of CTGAN to preserve correct categorical relations by 63%. Despite the enhancements the evaluation of the synthetic datasets shows that there is still room for improvement of the overall architecture because it seems difficult for the CTGAN model to efficiently learn all categorical constraints automatically.

Keywords: CTGAN · Categorical Relations · E-Commerce · *CV-deviation.*

1 Introduction

One of Amazon's biggest success factors has been its personalized recommendation system, which is based on massive amounts of data involving transactions from millions of customers [27]. Recommendation systems in general need a lot of data to learn the relationships between products, preferences and customer segments. Not only in the recommendation systems sector, but also in many other e-commerce areas, the analysis of customer data is a key element, e.g. in detection of fraud, in forecasting and inventory management. To solve the increasing demand new ways and models to create e-commerce data are urgently needed [17]. Due to its personal and sensitive nature, handling customer data brings its own challenges: How can people's personal data be protected, when it is used for analysis [7]? How can sensitive data be shared and multiplied? A promising solution to this problems is synthetic data: The generation of new data containing as many properties and information of the original data as possible while not being linked to the same individuals present in the original dataset [2]. Synthetic data research is an important area that is becoming increasingly popular throughout the machine learning field [18].

Especially in Europe, with the new data protection law GDPR, research achievements in synthetic e-commerce data are of central importance: The GDPR has led to a decline in usable data and seems to affect the revenues of the European e-commerce platforms [8]. The ability to use synthetic data to preserve information content while effectively protecting customer privacy could mean a breakthrough in European e-commerce market. Especially smaller players who are uncertain about the risks of using customer data, could benefit from the usage and sharing of synthetic data.

One architecture that is widely known due to numerous successes in generating so-called "deep fakes", i.e. deceptively real synthetic images, videos or audio content, is the Generative Adverisal Network (GAN) [24]. The Conditional Tabular Generative Network (CTGAN), a specialization of the GAN architecture for synthesizing tabular data was presented in 2019 by Lei Xu et al. in 'Modeling Tabular Data Using Conditional GAN' [26]. A first evaluation of the CTGAN in generating synthetic insurance data focusing on datasets with scalar values achieved promising results [14].

In this paper, we address the question of whether the CTGAN architecture can provide promising results in the area of synthetic e-commerce data. E-commerce data contains many columns with categorical data such as product categories or postal codes. The correlations between products and other columns with categorical data is central to recommender systems [20].

We evaluate synthetic e-commerce datasets along several dimensions and pay close attention to the maintenance of important relationships between the columns with categorical data. The similarity of categorical correlations between synthetic and original datasets is measured by the newly proposed measure Cramer's V Deviation (*CV-deviation*).

In our experiments, we generate synthetic data from a publicly available e-commerce dataset using CTGAN and implement a grid search that optimizes a subset of CTGAN's hyperparameters in respect to the e-commerce data. Additionally to evaluating a base approach, we include the *CV-deviation* measure in the hyperparameter training of the CTGAN model to investigate if this change of focus in the hyperparameter training has an effect on the evaluation parameters. By incorporating *CV-deviation* into the performance metric of the hyperparamter training, we can increase the the average categorical integrity of the synthetic e-commerce dataset by 17 percentage points.

The following Section 2 contains preliminaries on the CTGAN architecture. Subsequently, we present related work to this paper in Section 3. Section 4 displays our method including the formula for *CV-deviation* and introduces our implementation process. Section 5 discusses the evaluation of the synthetic e-commerce datasets. Lastly, Section 6 presents the central findings of this paper and points out further research directions.

2 Preliminaries

The GAN architecture was published in 2014 by Ian Goodfellow and his team, it consists of two artificial neural networks, the generator G and the discriminator D, which resemble two players playing a minmax game against each other [9]. The generator G produces synthetic data samples of a desired instance from a random noise source. Alternating with original samples from the real data distribution, these synthetic samples are fed into the discriminator network D, which determines whether the input belongs to the real dataset. During training the generator learns to create more realistic instances, while the discriminator tries to identify those generated instances with greater accuracy [9].

The CTGAN, whose architecture is illustrated in Figure 1, consists of two neural networks a generator G and a critic C that corresponds to the discriminator in the classic GAN architecture. The CTGAN's critic scores either 10 real or generated data series according to the network's estimated authenticity of the data. There are two main innovations adapted to the generation of synthetic tabular data: mode specific normalization and a conditional training by sampling [26].

Mode specific normalization is used to transfer the values of the continuous columns into a combination of a scalar value and a one-hot vector that increases the ability of CTGAN to create continuous columns with multiple modes during generation process. Figure 1 shows two datasets as input to the critic, one consisting of 10 rows of real data samples and the other consisting of 10 synthetic data rows. The continuous values of each row in those sets are represented using a mode-specific normalization and their categorical values are represented as one-hot vectors.

Another problem with GANs when creating tabular data is the highly imbalanced categorical columns, i.e., a column that consists of 90 % of one major category. CTGAN's conditional training approach applies a specific column with categorical data and a specific category from that column as a constraint to the data generation and the sampling process. Through an integration in the loss function, CTGAN learns to implement this condition during training. Figure 1 illustrates the influence of this condition as a conditional vector for the generator G and as a sampling filter for the input of the critic C. The column that is affected by this condition is chosen at random. To choose the category of the selected column as a condition, a probability mass function is calculated in which the calculated probability mass of each category is the logarithm of the frequency of that category in the selected column. To ensure that the infrequent

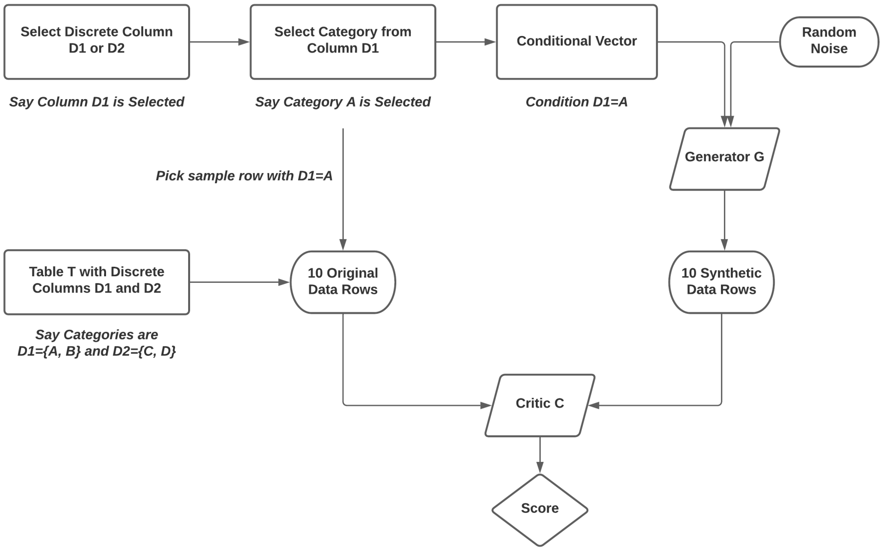

Select Discrete Column D1 or D2
Say Column D1 is Selected
Select Category from Column D1
Say Category A is Selected
Conditional Vector
Condition D1=A
Random Noise
Generator G
Pick sample row with D1=A
Table T with Discrete Columns D1 and D2
Say Categories are D1={A, B} and D2={C, D}
10 Original Data Rows
10 Synthetic Data Rows
Critic C
Score

Fig. 1: An illustration of CTGAN architecture.

categories of this selected column are considered to a larger extent, the category for the condition is randomly determined from this calculated probability mass function.

In addition to these enhancements, CTGAN also utilizes recent advances in GAN training such as WGAN-GP, an improved Wasserstein GAN with gradient penalty [10].

3 Related Work

Since its publication in autumn 2019, CTGAN has been combined with other architectures and evaluated with different datasets. Rosenblatt et.al. (2020) made the architecture differentially private, a formalization of privacy [6], by combining CTGAN with DP-SGD and PATE technique [19]. Similar to the DPGAN approach [25], applying DP-SGD to CTGAN adds random noise to the discriminator and prunes the norm to achieve differential privacy [19]. The PATE-CTGAN approach is also inspired by a similar approach on the GAN architecture [12], the original dataset is divided into subsets and each of the subsets has its own generator and discriminator network [19].

CTGAN has been evaluated on other types of table data and achieved promising results. Similar to our approach, Kuo et al. made small changes to the original CTGAN architecture that promise advantages for generating insurance datasets [14]. Their workflow involves using true data frequency instead of log-frequency when choosing a value for the condition that influences the generator and the sampling process. In 2021, Min Jong et.al. evaluated the CTGAN and the TGAN in their ability to generate synthetic EEG data. In their evaluations, the CTGAN achieved higher similarity scores than the TGAN, while the machine learning performance of the two generative architectures remained similar [15]. In order to obtain better training data for the evaluation of stability of power systems, Han et al. use an approach that first creates tabular data with CTGAN, which is then further processed. The data created with this framework is analyzed with different methods and achieves good values in several metrics [11]. None of the above CTGAN evaluations focus especially on relationships between columns with categorical data, which are in particular important in e-commerce data where, for example, products are divided into different categories that must be preserved in the synthetic data.

4 Dataset

For our synthesis of e-commerce data, we use the "Superstore" dataset that contains data on purchases from 2014 to 2017 from an U.S. online store with various offerings ranging from books to furniture or other household items. To increase transparency, we choose a dataset which is publicly available on the Kaggle platform. The dataset does not appear to be anonymized [23].

The unprocessed "Superstore" dataset consists of 9,994 rows and 20 columns with different information. In order to use the "Superstore" dataset for synthesis, some information duplicated with the respective ID such as *Product Name* and *Customer Name* are deleted. The "Superstore" dataset has 13 columns with categorical data, the remaining 4 columns are continuous. The dataset includes 793 individual customers who ordered 1862 different products in 5,009 individual orders. All products can be divided into 17 subcategories, which in turn are divided into the three categories *Furniture*, *Office Supplies* and *Technology* [23].

5 Method and Implementation

In this section, we describe our method and the steps in our implementation. We start by introducing *CV-deviation*, our new evaluation metric for synthetic datasets to better incorporate columns with categorical data. We then display the evaluation methods we use to analyze the synthetic e-commerce datasets. Furthermore, we describe the general implementation of training the CTGAN models. Lastly, we present our technical setup and performed grid search.

5.1 Cramer's V Deviation

In order to achieve higher similarity between synthetic and original e-commerce data, we intend to use a performance metric that especially supports categorical integrity of synthetic data. The Cramer's V is a measure of statistical association between two categorical variables, it returns a value between 0 and 1, with a higher value representing a greater correlation of the two variables. The Cramer's V with Wicher Bergsma correction (CV) for column pair (D_i, D_j) with categorical data and number of categories $|D_i|$ and $|D_j|$ in table T with Number of rows N_r is calculated as follows [4]:

$$\begin{gathered} CV = \sqrt{\frac{\tilde{\Phi}^2}{min(\tilde{k}-1, \tilde{r}-1)}}, \\ \tilde{\Phi}^2 = max(0, \Phi^2 - \frac{(k-1)(r-1)}{(n-1)}),\ \Phi^2 = \frac{\chi^2}{n}, \\ \tilde{k} = k - \frac{(k-1)^2}{n-1},\ \tilde{r} = r - \frac{(r-1)^2}{n-1}, \\ \chi^2 = \text{chi-square test of independence [13] of } (D_i, D_j), \\ k = |D_i|,\ r = |D_j|,\ n = N_r \end{gathered} \tag{1}$$

We define the corrected Wicher Bergsma Cramer's V of a column pair with categorical data $j \in \mathcal{P}_2(D_1, ..., D_{N_d})$ of tabular dataset T with columns with categorical data $D = \{D_1, ..., D_{N_d}\}$ as: $CV_T(j)$.

To create a performance metric using CV as a base, we combine the Cramer's V with Wicher Bergsma correction with the Root Mean Squared Error [5] and obtain the Cramer's V Deviation (*CV-deviation*). The *CV-deviation* of real table T and synthetic table T_{syn} with columns with categorical data $D = \{D_1, ..., D_{N_d}\}$ and $(|D| > 1)$ is calculated as follows:

$$CV\text{-}deviation(T, T_{syn}) = \sqrt{\frac{1}{|\mathcal{P}_2(D)|} \sum_{j \in \mathcal{P}_2(D)} (CV_T(j) - CV_{T_{syn}}(j))^2} \tag{2}$$

The *CV-deviation* measures the difference of the statistical correlations between all pairs from the columns with categorical data in the synthetic table T_{syn} to the corresponding correlations in the real table T. The *CV-deviation* is a similarity measure for a pair of real and synthetic tabular data and therefore can only be applied to datasets that have the same columns with categorical data. The *CV-deviation* can take 0 as the lowest result and 1 as the highest value, the closer the result is to 0, the more similar are the statistical relationships among the columns with categorical data with respect to the CV value. If we were to calculate the *CV-deviation* from a dataset to itself, the result would be 0.

5.2 Evaluation Method

We evaluate the synthesized e-commerce datasets in detail and compare them with the original dataset. Therefore, we examine the similarity of column distributions of synthetic datasets to the column distributions of the original "Superstore" dataset looking at overall distribution measures, upper and lower bounds for continuous columns and number of categories for columns with categorical data. We also inspect the integrity of relationships between column pairs with categorical data in our synthetic datasets. Some columns with categorical data in e-commerce data have a special relationship to each other that does not allow new combinations in the synthetic data, i.e., a cellphone always belongs to the sub-category technology and not furniture. Detection of column pairs with this categorical integrity requires expert knowledge about the relationships between columns, which is not always available. We furthermore compare the Cramer's V values of the synthetic datasets with the results of the original dataset to display an overview of the similarity of categorical statistical correlations of the synthetic datasets to the original dataset.

5.3 General Implementation

The CTGAN training process and all evaluation of the synthetic datasets is written in Python 3.7. We create CTGAN models using version 0.12 of the Synthetic Data Vault (SDV) library. The SDV library is an set of open source software systems concerning synthetic data, this project was launched by the Massachusetts Institute of Technology in 2018 [21]. The implementation of the SDV CTGAN is the realization of the original paper [26]. For each combination of the selected hyperparameters that we optimize in this paper, we create a CTGAN model that is trained with the appropriate parameters on the "Superstore" dataset. After training, we save every CTGAN model and create 10,000 rows of synthetic data with the saved model.

5.4 Gridsearch and Technical Setup

For optimizing CTGAN in respect to e-commerce data we implement a grid search over the following hyperparameters: epochs {100, 300, 500, 700, 900}, batch size {100, 300, 500, 700, 900, 1000}, log frequency {True, False} (whether to use the logarithm of the frequency of a value in a column with categorical data to determine the conditional input), learning rate (LR) for the critic {2e-4, 2e-5} and critic steps {1,5} (number of critic updates to do for each generator update). The original CTGAN uses 1 critic update and the default from the WGAN-GP paper is 5 [10]. The grid search results in 240 different CTGAN models. We compute each model on a Quadro RTX 6000 and parallelize this process multiple times on a cluster server.

Each of the 240 created synthetic datasets is evaluated with two performance metrics. The first performance metric is the SDV Single Table Metric, which is included in the SDV library. The SDV Single Table Metric itself is a collection of other lower level metrics that can be divided into multiple groups. We use the following three groups of metrics (based on the SDV framework) to measure the quality of our synthetic data:

statistical metrics: KSTest, CSTest
likelihood metrics: GMlikelihood
detection metrics: LogisticDetection

The SDV Metric returns a score between 0 and 1, being 0 the worst and 1 the best possible score [21].

As a second performance metric, we supplement the SDV metric with an additional component for categorical relationships: we use a combined and equally weighted score from the normalized SDV metric value and the normalized 1 - CV deviation value. This combined metric, CVSDV, also scores the synthesized datasets on a scale of 0 to 1, with a score closer to 1 indicating higher similarity to the original dataset.

6 Results and Discussion

In this chapter, we evaluate two synthetic datasets, the dataset that scores highest in the SDV metric and the dataset that has the highest score in our new CVSDV metric. We start by evaluating the similarity of the synthetic column distributions to the original distributions. Afterwards, we inspect how closely the original relationships of the columns with categorical data are transferred to the synthetic data. The two best combination of hyperparameters in terms of SDV metric and CVSDV metric are shown in Table 1.

Table 1: Hyperparameters synthetic datasets with highest SDV or CVSDV

Dataset	Epochs	Batch Size	Log Freq.	Cr. Steps	Cr. LR	SDV	CVSDV
highest SDV	100	1000	False	5	2e-4	0.6323	0.6477
highest CVSDV	500	900	False	5	2e-4	0.5835	0.7945

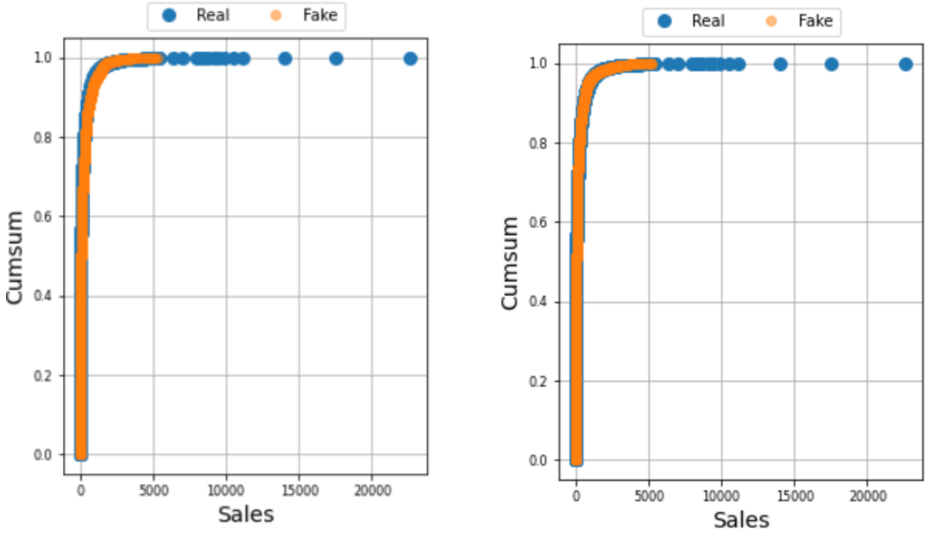
6.1 Column Distribution

Both the Kolmogorov–Smirnov test [3] for continuous columns and the Chi-Squared test [13] for columns with categorical data show high similarity between synthetic and original dataset (see Table 2).

Table 2: Chi-Squared test and Kolmogorov–Smirnov test results.

Synthetic Dataset	Chi-Squared Test	Kolmogorov–Smirnov test
highest SDV score	0.998794	0.880171
highest CVSDV score	0.995688	0.908547

All continuous columns of the two synthetic datasets do not exceed the value range of the original columns. However, it is noticeable that both synthetic datasets strongly decimate the upper and lower limit of the value range in some continuous columns like *Sales*. The table-evaluator [22] Figure 2 shows the synthetic datasets cumsum of the *Sales* column, a statistical quality control that



(a) Dataset highest SDV score.

(b) Dataset highest CVSDV score.

Fig. 2: Cummulative sum *Sales* column in synthetic datasets.

measures change [1]. We recognize a few purchases of more than 5,000$ in the *Sales* distribution of the original dataset, these edge cases are missing in both synthetic datasets resulting in a about 76% smaller value range. Overall both synthetic datasets lack about 40% value range in continuous columns compared to the original dataset.

Some synthetic columns have fewer categories than the columns in the original dataset. For example, this is the case for columns like *Product ID* that have a large amount of possible categories (1862). Columns with fewer categories like *Region* or *Sub-Category* contain the same categories as the original dataset. Figures 3, 4a and 4b display the distribution of the column with categorical data *Sub-Category*: a strong increase of purchases of “Bookcases” can be seen in both synthetic datasets. The cause for these unusually high “Bookcases” values could be mode collapse, a failure typical for GAN architectures [26].

6.2 Categorical Integrity

Figures 5, 6a and 6b illustrate the Wicher Bergsma Cramer’s V (CV) [4] values of all pairs of columns with categorical and temporal data in the original dataset and in the two synthetic datasets. In the heatmaps, a high CV score indicating high statistical association is connected with a lighter color. The original dataset has high CV values between columns in the lower left quarter of the heatmap, this pattern is more evident in the heatmap of the dataset whose hyperparameters were optimized with the CVSDV metric. The best SDV score dataset has overall

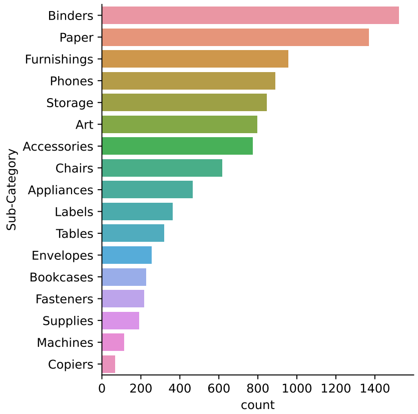


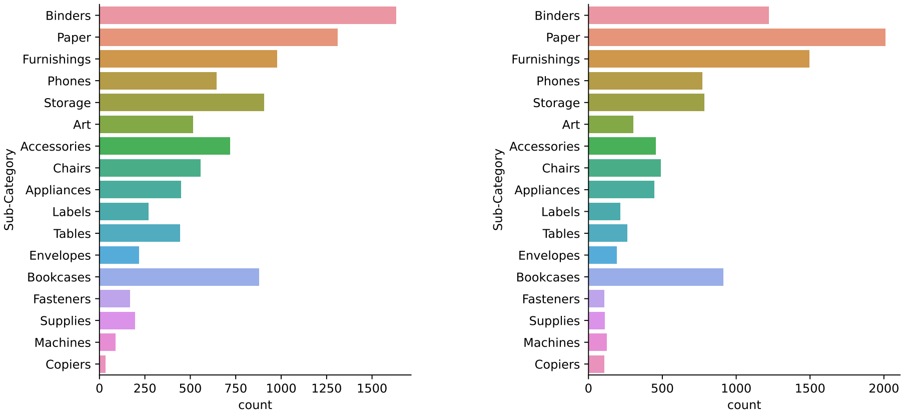
Fig. 3: Distribution *Sub-Category* column original dataset.

much lower CV values, visualized by the lower maximum value of the scale of the heatmap at 0.25, compared to the maximum value of the scale of the CVSDV dataset heatmap, which reaches a value of 0.6. The *CV-deviation* value confirms a closer proximity of the best CVSDV dataset to the original dataset, the *CV-deviation* of the best CVSDV dataset is 0.32 and that of the best SDV dataset reaches a higher value of 0.37.

The original CV heatmap (Figure 5) visualizes some special categorical relationships that do not allow new combinations in the synthetic dataset with very high values, such as *City* and *Postal Code* with a CV value of 0.99. In Table 3 we can see the absolute numbers and percentage by which both synthetic datasets correctly reflect these type of relationships. The synthetic dataset, whose model was optimized with the CVSDV metric, achieves a higher number of correct matches for each individual column pair. The largest absolute difference occurs at (*State*/*Region*), here the CVSDV dataset achieves a better result by

Table 3: Categorial integrity in synthetic datasets.

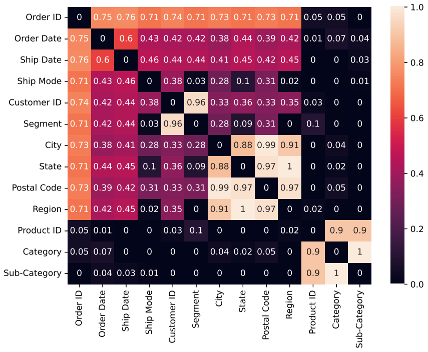
Column Pair	Number Combinations	Correct SDV	Correct CVSDV
(*Category*/*Sub-Category*)	17	5,315(53%)	7,865(79%)
(*Category*/*Product ID*)	1,862	4,385(44%)	4,442(44%)
(*Product ID*/*Sub-Category*)	1,862	864(9%)	1,007(10%)
(*City*/*State*)	604	1,268(13%)	3,099(31%)
(*City*/*Postal Code*)	632	489(5%)	1,643(16%)
(*City*/*Region*)	583	4,124(41%)	6,506(65%)
(*State*/*Postal Code*)	631	1,015(10%)	2,546(25%)
(*State*/*Region*)	49	3,589(36%)	6,968(70%)
(*Region*/*Postal Code*)	631	2,982(30%)	5,226(52%)



(a) Dataset highest SDV score.

(b) Dataset highest CVSDV score.

Fig. 4: Distribution *Sub-Category* column in synthetic datasets.

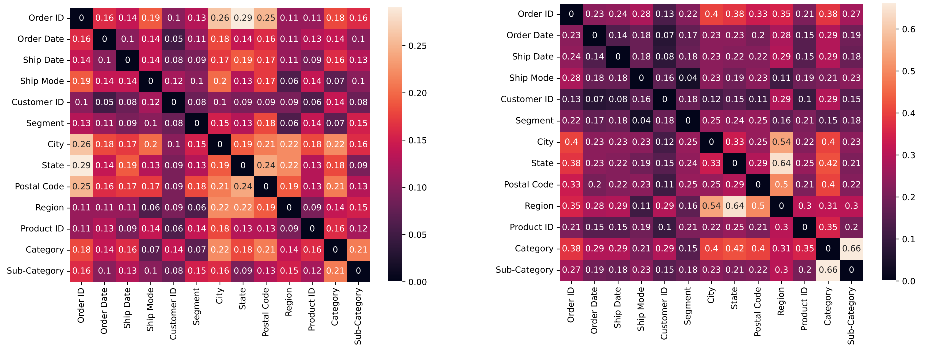


	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Segment	City	State	Postal Code	Region	Product ID	Category	Sub-Category
Order ID	0	0.75	0.76	0.71	0.74	0.71	0.73	0.71	0.73	0.71	0.05	0.05	0
Order Date	0.75	0	0.6	0.43	0.42	0.42	0.38	0.44	0.39	0.42	0.01	0.07	0.04
Ship Date	0.76	0.6	0	0.46	0.44	0.44	0.41	0.45	0.42	0.45	0	0	0.03
Ship Mode	0.71	0.43	0.46	0	0.38	0.03	0.28	0.1	0.31	0.02	0	0	0.01
Customer ID	0.74	0.42	0.44	0.38	0	0.96	0.33	0.36	0.33	0.35	0.03	0	0
Segment	0.71	0.42	0.44	0.03	0.96	0	0.28	0.09	0.31	0	0.1	0	0
City	0.73	0.38	0.41	0.28	0.33	0.28	0	0.88	0.99	0.91	0	0.04	0
State	0.71	0.44	0.45	0.1	0.36	0.09	0.88	0	0.97	1	0	0.02	0
Postal Code	0.73	0.39	0.42	0.31	0.33	0.31	0.99	0.97	0	0.97	0	0.05	0
Region	0.71	0.42	0.45	0.02	0.35	0	0.91	1	0.97	0	0.02	0	0
Product ID	0.05	0.01	0	0	0.03	0.1	0	0	0	0.02	0	0.9	0.9
Category	0.05	0.07	0	0	0	0	0.04	0.02	0.05	0	0.9	0	1
Sub-Category	0	0.04	0.03	0.01	0	0	0	0	0	0	0.9	1	0

Fig. 5: Cramer's V of column pairs in original dataset.

3379 correct data rows, which means an increase of 94 % compared to the SDV dataset. Applying the CVSDV performance metric increases the number of correctly assigned rows for the column pair (*City*/*Postal Code*) by as much as 235%, which is the highest percentage improvement. It is noticeable that for categories with many possible combinations like (*Product ID*/*Sub-Category*) both synthetic datasets achieve very low matches. Overall, the SDV dataset achieves an average of 27 % correct matches and the CVSDV dataset average is 17 percentage points (or 63%) higher at 44 % correct categorical assignments.

For completeness, we briefly consider the temporal integrity of the synthetic datasets. There are also temporal requirements that must be met in the synthetic data in order to reflect a real purchase process, e.g. the *Ship Date* must be temporally after the *Order Date*. The correct chronological order of the date columns



(a) Dataset highest SDV score.

(b) Dataset highest CVSDV score.

Fig. 6: Cramer's V of column pairs in synthetic datasets.

is maintained in both datasets to 50 percent, the CVSDV dataset reaches only minimally better values than the SDV dataset.

7 Conclusion

Overall, the CTGAN architecture seems to be a promising architecture to generate e-commerce data. Both synthetic datasets have similar column distributions to the original dataset and the reduction of the definition range in continuous columns only plays a minor role, since this only affects a small subset of the data.

For a real world application of synthetic e-commerce data, it is important that each data row reflects a correct buying processes, and therefore keeping correct categorical relationships is a key point. For both evaluated synthetic datasets, this categorical integrity is only maintained at an average percentage of less than 50 percent: SDV metric dataset (27%) and CVSDV metric dataset (44%), which is not satisfactory. Especially for column pairs with a large number of categories, CTGAN has problems to reflect their relationships correctly in the synthetic data. However, there is a significant overall increase in the dataset whose CTGAN hyperparameters are optimized with the CVSDV metric. Applying the CVSDV performance metric more than doubles the number of correct assignments for some column pairs and improves the average categorical integrity by 17 percentage points.

In order to use CTGAN for the production of synthetic e-commerce data, other approaches are still needed that will lead to better categorical integrity. One approach, could be to integrate statistical evaluation metrics, such as the presented *CV-deviation*, into the direct training process of CTGAN and thus enforce greater adherence to categorical relations at an earlier stage. Another interesting approach could be increasing the pac size, i.e., the number of data rows that the critic receives as samples, to more than 10. Viewing multiple rows of data simultaneously could make the correlations between columns more visible to the network and improve the ability of the CTGAN architecture to translate such relationships into synthetic data. To improve the overall performance of the CTGAN architecture other loss function could be tested which lead to a good performance in current GAN models like the adversial loss used in NSGAN with R1 regularization [16].

Another research direction would be to add the training of a real-life recommender system as a subsequent evaluation step. The recommender performance achieved with the synthetic dataset could be then compared with the recommender perforamance of the original dataset.

References

1. Barnard, G.A.: Control charts and stochastic processes. Journal of the Royal Statistical Society: Series B (Methodological) **21**(2), 239–257 (1959)
2. Bellovin, S.M., Dutta, P.K., Reitinger, N.: Privacy and synthetic datasets. Stan. Tech. L. Rev. **22**, 1 (2019)
3. Berger, V.W., Zhou, Y.: Kolmogorov–smirnov test: Overview. Wiley statsref: Statistics reference online (2014)
4. Bergsma, W.: A bias-correction for cramér's v and tschuprow's t. Journal of the Korean Statistical Society **42**(3), 323–328 (2013)
5. Chai, T., Draxler, R.R.: Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific model development **7**(3), 1247–1250 (2014)
6. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008)
7. Gahi, Y., Guennoun, M., Mouftah, H.T.: Big data analytics: Security and privacy challenges. In: 2016 IEEE Symposium on Computers and Communication (ISCC). pp. 952–957 (2016). https://doi.org/10.1109/ISCC.2016.7543859
8. Goldberg, S., Johnson, G., Shriver, S.: Regulating privacy online: The early impact of the gdpr on european web traffic & e-commerce outcomes. Available at SSRN 3421731 (2019)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS (2017)
11. Han, G., Liu, S., Chen, K., Yu, N., Feng, Z., Song, M.: Imbalanced sample generation and evaluation for power system transient stability using ctgan. In: International Conference on Intelligent Computing & Optimization. pp. 555–565. Springer (2021)
12. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations (2018)
13. Koch, G.G., Wiener, L.E.: Chi-squared tests: Basics. Wiley StatsRef: Statistics Reference Online pp. 1–20 (2014)
14. Kuo, K., et al.: Generative synthesis of insurance datasets. Tech. rep. (2020)

15. Lee, J.S., Lee, O.: Ctgan vs tgan? which one is more suitable for generating synthetic eeg data. Journal of Theoretical and Applied Information Technology **99**(10) (2021)
16. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018)
17. Moorthi, K., Dhiman, G., Arulprakash, P., Suresh, C., Srihari, K.: A survey on impact of data analytics techniques in e-commerce. Materials Today: Proceedings (2021)
18. Nikolenko, S.: Synthetic data in deep learning. In: School-conference "Approximation and Data Analysis 2019". p. 21 (2019)
19. Rosenblatt, L., Liu, X., Pouyanfar, S., de Leon, E., Desai, A., Allen, J.: Differentially private synthetic data: Applied evaluations and enhancements (2020)
20. Sarker, I.H.: Machine learning: Algorithms, real-world applications and research directions. SN Computer Science **2**(3), 1–21 (2021)
21. Sdv - the synthetic data vault — sdv 0.12.1 documentation. https://sdv.dev/SDV/, accessed: 2021-8-13
22. table-evaluator: A package to evaluate how close a synthetic data set is to real data. https://pypi.org/project/table-evaluator/, accessed: 2021-12-13
23. Us superstore data. https://www.kaggle.com/juhi1994/superstore, accessed: 2021-09-30
24. Whittaker, L., Kietzmann, T.C., Kietzmann, J., Dabirian, A.: "all around me are synthetic faces": The mad world of ai-generated media. IT Professional **22**(5), 90–99 (2020)
25. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. arXiv e-prints pp. arXiv–1802 (2018)
26. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. Advances in Neural Information Processing Systems **32**, 7335–7345 (2019)
27. Zakir, J., Seymour, T., Berg, K.: Big data analytics. Issues in Information Systems **16**(2) (2015)