Comparison of Synthetic Data Generation Tools Using Internet of Things Data

Bachelor Project Information Sciences

Darin Pavlov

Supervisor: Roderick van der Weerdt

Vrije Universiteit, Amsterdam, The Netherlands

July 2022

Abstract. The Internet of Things (IoT) is a complex infrastructure of interconnected smart devices. Within that infrastructure, large volumes of data are constantly being collected and transmitted through sensors. In combination with other innovative technologies such as Machine Learning, IoT finds wide application in a number of fields such as automotive, healthcare, manufacturing, etc. However, research in the area is hindered, since the required IoT data often contains sensitive information, which reflects on its availability. A solution for this problem is to use synthetic data, which resembles as much as possible the real one. In this study, we conduct an experiment, in which we investigate the effectiveness of synthetic IoT data generation by three different tools, namely Mostly AI, Gretel.ai, and SDV, and compare their utilities, based on statistical and distinguishability metrics. We observe that Mostly AI outperforms the other two generators, although Gretel.ai shows similar satisfactory results on the statistical metrics. The output of SDV on the other hand is poor on all metrics. Through this study we aim to encourage future research within the quickly developing area of synthetic data generation in the context of IoT technology.

Keywords: Synthetic Data, Internet of Things, Synthetic Data Generators, Synthetic Data Utility

1 Introduction

The Internet of Things (IoT) is a promising technology, which comprises a global network of smart devices, growing in numbers by the day [1]. It is based on several

protocols and cutting-edge technologies, such as Radio Frequency Identification, which enables sensing objects through tags and readers, and Wireless Sensor Networks. Innovative companies produce more and more devices that can be connected to the Internet and to other devices. IoT is increasingly gaining relevance across many industries, including healthcare, automotive, security, and agriculture. Its most valuable feature is the interconnected nature of the devices, which allows for many applications, such as smart agriculture, domestic and home automation, smart cities, and many more.

In the healthcare industry, IoT technology is envisioned to dynamically track and monitor patients' health data [2]. It is closely related to wearable technologies, as they involve many sensors, which allow for better interaction with the external environment. For instance, [3] suggests an IoT-based solution for asthma. The respiratory rate of patients is recorded by a smart sensor that measures the temperature of inhaled and exhaled air. The sensor is connected to an alarm that is triggered in case of a detected anomaly.

Another popular application of IoT is in Smart Home systems [1]. Modern households use smart electronic devices, equipped with sensors and actuators, which generate and exchange data. In combination with powerful Machine Learning capabilities, the IoT network enables home automation, adapted to users' needs. For example, it can regulate the room temperature and energy consumption, it can detect intruders, and so on.

Although the IoT gains popularity, research in the area faces some challenges [4]. As mentioned before, the connection of smart devices relies on infrastructure that enhances traffic of big volumes of data. This leads to two main problems. First, the size of the data is crucial for the training and validation of Machine Learning models, so that they can find meaningful patterns, using real-life data. However, generating enough data to achieve this is time-consuming and therefore restricts the training and validation processes. Second, the large volumes of data are often considered of sensitive nature. The main concerns are related to the way personal data is gathered, managed, exploited, and secured. As a consequence, access to such data is limited, which complicates the research and development.

A common solution to the aforementioned problems is to use synthetic data, which highly resembles real-life data [5]. There are various benefits of using synthetic data. For example, [5] addresses two of them: enhancement of analytics and easier access to data. The authors argue that often data accessibility is limited, and data scientists have to rely mainly on open-source data, that might not fit their research area. They also stress on the importance of synthetic data, when the data that is required does not exist, as its collection is unethical, impractical, or difficult.

Finding a reliable method to reproduce IoT data, is expected to aid researchers in conducting their studies in the IoT domain, without any privacy or accessibility concerns. This study focuses on exploring and comparing how current methods for generating synthetic data perform when the task is to generate IoT data. The research question that the paper aims to answer is, therefore: **Can we create credible synthetic IoT datasets using pre-existing dataset generators?** We answer this question by conducting an experiment in which synthetic IoT datasets are generated and compared

based on the level of resemblance to their original counterparts. The resemblance is measured on two metrics of synthetic data utility- statistical resemblance and indistinguishability. In addition, the statistical significance of the differences between the synthetic datasets is verified through an Analysis of Variance (ANOVA) test.

The next sections of this paper are structured as follows: Section 2 discusses related work and the contribution of this study to the research in the field. Section 3 provides a detailed description of the set-up of the experiment, by elaborating on the involved datasets, generators, and evaluation method. The results of the experiment are presented in Section 4. Finally, Section 5 discusses the outcomes and the limitations, as well as outlines the possibilities for future research.

2 Related work

2.1 Application of Synthetic Data for IoT

In the context of IoT, synthetic data takes place in a number of applications. In this section, we describe two relevant examples, which show the value that synthetic data can bring to IoT technology.

The authors of [6] have suggested a benchmark tool- IoTAbench, which is built for testing Big Data in IoT use cases. The solution helps to understand and analyze how the scenarios will work. To achieve this, the benchmark uses a synthetic data generator, based on an augmented Markov chain model, to generate time series data that imitate said scenarios. The Markov-chain-based methods have been considered appropriate for the purposes of this work, as they manage to model sequences successfully. However, we could not find an accessible and suitable solution, which motivated us to investigate other approaches.

Another paper [7] describes a data generation method for synthesizing smart grid time series data. The authors argue that there is a lack of such data, which hinders research for developing Machine Learning solutions for smart grid optimization. They evaluate the synthetic data by performing Machine Learning tasks, such as predicting energy consumption for the next 24 hours and clustering users based on their consumption. After observing the satisfactory results, the authors conclude that the synthetic data is indistinguishable from the original. [7] shows how Machine Learning can be used to test the utility of the data, which influences the choice of metrics for the experiment in this paper.

2.2 Generation of Synthetic IoT Data

Several solutions for generating synthetic IoT data have been developed. As outlined in [8], in healthcare, the need for synthetic data is increasing as real-life data contains sensitive information about patients and the access to it is often limited. The authors also argue that data used to train and validate Machine Learning models in the healthcare sector needs to be as realistic and complex as possible. In their work, they present a synthetic data generating system, called SynSun, which is based on hidden

Markov models and regression models, trained on a real-life dataset. They conclude that the SynSun helps to produce better activity recognition accuracy than a real dataset. Although the suggested generator to be tested in the experiment of this study, its documentation is outdated, and its code does not work properly. In addition, it requires a specific classification of the input data, which could not be done with the tested datasets.

Another example of a synthetic data tool from the healthcare domain is the Advanced Patient Data Generator (APDG) [9]. What is special about this case is that generation of data is controlled through domain knowledge, collected from biomedical publications and further formalized in the Patient Data Definition Language (PDDL). The method is applied to generate data for breast cancer patients. The authors argue that the domain knowledge approach leads to more realistic results. However, this generator is not accessible, thus it was not possible to include it in this research.

The limitations of the explored existing synthetic IoT data generators have inspired this work to investigate solutions outside of the IoT domain. This is why the tools that have been tested are not specifically intended for synthesizing IoT data.

2.3 **Synthetic Data Utility**

Synthetic data utility is a term to describe how much the synthetic data resembles real data, given a task that it has to complete. However, there are multiple ways to measure it. [10 below] explores how to optimize the utility of data based on its application. The authors argue that data utility can have multiple dimensions, and the metrics by which it can be determined highly depend on the use case. Seven main use cases are outlined and for each, the authors describe how data utility can be optimized through synthetic data. One of the use cases is Machine Learning, in which they describe three applications of synthetic data. The first one is evaluating and comparing different Machine Learning algorithms. The authors claim that data can be generated for the training, validation, and testing of algorithms regardless of the size of the original dataset. The second application is data augmentation, which prevents class imbalances in datasets. This problem results in poor performance of the algorithm on the underrepresented class. Finally, [10] argue that synthetic data can find application in preserving the privacy of data that is used for Machine Learning algorithms, as recovering the training data is becoming easier nowadays. However, if the training data is synthetic, the risk of uncovering sensitive data is reduced. Furthermore, the authors define six metrics to evaluate the quality of the synthetic data - Hellinger distance, prediction accuracy, bivariate correlation, area under the receiver operating characteristics, distribution comparison, and distinguishability. Our study compares synthetic data generators based on the last two metrics.

In [11] a similar study is conducted, in which synthetic data generators are compared. Their work is focused on another metric of the synthetic data utility - the efficacy of Machine Learning models, which are trained on the synthetic data. In contrast, in our study, we investigate the statistical and distinguishability metrics of

different generators. In addition, we test the performance of the generators exclusively on IoT datasets.

3 Methodology

We have conducted an experiment that aims to compare the utilities of three different data generators, namely Mostly AI [12], Gretel.ai [13], and Synthetic Data Vault (SDV) [14]. The experiment consists of producing synthetic datasets from two real IoT datasets through the three generators. The outcomes are then evaluated using two types of metrics of utility -statistical and detection. To define satisfactory results, we have also evaluated subsets of the original datasets on the same metrics, to serve as a baseline for the results of the generators. Moreover, as the synthetic datasets are expected to be as close as possible to the original ones, we hypothesize that there is no difference between the means of the three generators. This is verified through an ANOVA test. If we find that the utilities of all synthetic datasets are satisfactory and fail to reject the hypothesis, we can positively answer the research question. Further in this section, we describe the setup of the experiment in detail.

3.1 Datasets

Finding a suitable IoT dataset has proven to be challenging, as most such datasets are not easily accessible. Nevertheless, two datasets have been used for this experiment.

The first one contains household data, from the Open Power System Data website [15]. It measures the total electricity consumption of different devices in several residential and industrial facilities. It is structured as time series with gaps of 60 minutes between timestamps. The values of the original dataset are the cumulative electricity consumption up until the corresponding timestamp. However, the cumulative nature of the data results in an additional factor to be taken into consideration and the generators failed to preserve such inter-row dependency. Therefore, each entry from the measurement columns is converted into the difference between itself and the previous measurement. Moreover, the dataset contains a lot of missing values. The dataset is valuable for the experiment, as it can help to demonstrate how well the generators work with realistic IoT datasets. It contains more than 8500 rows and 70 columns, which represent the consumption by different facilities and devices.

	utc_timestamp	DE_KN_industrial1_grid_import	DE_KN_industrial1_pv_1	DE_KN_industrial1_pv_2	DE_KN_industrial2_grid_import
0	2014-12-11 17:00:00	NaN	NaN	NaN	NaN
1	2014-12-11 18:00:00	NaN	NaN	NaN	NaN
2	2014-12-11 19:00:00	NaN	NaN	NaN	NaN

Fig. 1. Dataset 1

The second dataset is taken from Kaggle.com and contains time series data about electricity consumption in a smart home [16]. Unlike the first dataset, this one does not contain any missing entries. The readings have a span of 1 minute and the data has been collected for 365 days, which makes more than 500000 rows. However, only the first 20000 are taken for the experiment due to technical and time constraints. The dataset contains 20 columns out of which one is for the timestamp and the rest are measurements from different appliances of the smart home.

	time	use	gen	House overall	Dishwasher	Furnace 1	Furnace 2	Home office	Fridge	ine cellar	Garage door	Kitchen 12	Kitchen 14	Kitchen 38
0	2016-01-01 05:00:00	0.9328	0.0035	0.9328	0.0000	0.0207	0.0619	0.4426	0.1242	0.0070	0.0131	0.0004	0.0001	0.0
1	2016-01-01 05:00:01	0.9343	0.0035	0.9343	0.0000	0.0207	0.0638	0.4441	0.1240	0.0070	0.0131	0.0004	0.0001	0.0

Fig. 2. Dataset 2

3.2 **Generators**

Although many methods for generating synthetic data have been considered, the experiment has been focused on three of them.

Mostly AI. The first generator that has been tested is the Mostly AI synthetic data generator, which is provided through an online platform. Although the platform is not specifically intended for IoT data, the service is provided both for community and enterprise purposes with the former being free and therefore used for the experiment. One of the advantages of this tool is the interface, which ensures a straightforward process of generating data. Firstly, the datasets are pre-processed according to the predefined requirements, such as filling missing values with empty strings and changing the DateTime values to a specific format. Once the datasets are loaded as a CSV file, the columns to be used are selected, as well as their data types (although when the dataset is loaded, the tool tries to predict them) and some additional parameters, such as the granularity of the values and the number of processed and generated subjects. The next step is the tuning of model parameters, which are set as suggested by the developers - a maximum of 200 training epochs with a batch size of 32 and a learning rate of 0.001. The rest of the process is done automatically by the tool. First, there is an encoding step in which numerical and DateTime values are range limited between the 10th lowest and the 10th highest values of the original distribution. This is done so that no extreme values show up in the training data and consequently in the synthetic data. After that, the data is tokenized, split into training and validation sets, and passed on to the training step. The training step starts with adjusting the architecture of a Machine Learning model to the input and vectorizing the tokens. Then the model learns conditional probabilities in the feature space of independent data points, as well as in the feature space of sequences. The generation of new data

happens as the algorithm samples the learned probabilities of the trained model and detokenizes the output.

Gretel.ai. Gretel.ai (hereby referred to as Gretel) is an open-source platform and similarly to Mostly AI provides an interface, which facilitates the process of generating data with little technical knowledge required. In addition to creating synthetic data, the tool can discover and label sensitive data types on one hand and perform privacy-preserving transformations on the other. Although the range of possible inputs is wide, simple formats such as CSV are recommended. After the original dataset has been uploaded, the user can choose the configuration of the Machine Learning model. Gretel uses a long short-term memory neural network, which allows for better reproduction of sequential data, and has pre-built models that can be applied depending on the content and structure of the dataset. Thus, for this experiment, the chosen configuration is suitable for mainly numeric data – 100 epochs with a learning rate of 0.001 and 256 recurrent neural network units. A helpful feature of the tool is that users can use their own model configurations that better fit their needs. The automatic process starts once the configuration is set. The data is profiled and clustered and field-level statistics are extracted to be later used for validation of the generated data. Similar to Mostly AI, the data is tokenized and vectorized prior to training. The trained model can be reused to generate synthetic data records.

Synthetic Data Vault. SDV is another open-source tool that provides multiple models to synthesize data. For the purposes of this experiment, we have chosen a model that deals specifically with time series data – the probabilistic autoregressive (PAR) model. This generator does not require any specific preprocessing of the data. Instead, it focuses on labeling the columns correctly – as context columns, sequence index, or entity columns. The labeling helps the model determine and reproduce the inter-row and inter-column relationships in the datasets. Setting the time column as sequence index is the only necessary preprocessing, as other labels do not match our data. The model is then fit to the data and ready to generate synthetic data.

3.3 **Evaluation**

The evaluation includes two parts. Firstly, we check the statistical significance of the differences between the synthetic datasets through an ANOVA test. The test is applied with a significance level of 0.05. Equal samples are taken from each dataset and added to a new dataset where they are labeled according to their origin. An ordinary least squares model is then fitted to the new dataset. Finally, the model is passed to an ANOVA method, which essentially performs the test. This process is done in Python programming language by using the Statsmodels package [17].

Secondly, we test the utility of the synthetic datasets. The metrics that have been applied to do so are part of a framework provided by the SVD. The framework only takes as parameters the real dataset and its synthetic counterpart and provides a number

of evaluation metrics, such as Logistic Regression Detection, SVC Detection, Gaussian Mixture Log Likelihood, Chi-Squared, Kolmogorov-Smirnov statistic, continuous Kullback–Leibler divergence and others. The returned output is normalized to take values between 0 and 1. To define satisfactory results of the synthetic outputs, we take a sample from each of the original datasets and test it on the framework. We argue that if the results of the synthetic data are close to those of the original one, the three generators are capable of replicating the IoT data, although they are not intended for it. In addition, we set the tolerance for the difference between the results of the two types of data to 0.05. Our experiment measures the statistical resemblance, as well as the degree of difficulty to distinguish synthetic from real data.

Statistical Metrics. Two statistical metrics are applied for the evaluation – the Kolmogorov-Smirnov test and Continuous Kullback–Leibler divergence. The former measures the probability of two sets of samples taken from the real and the synthetic dataset belonging to the same distribution. This is done for every corresponding pair of columns - synthetic and real. The Kullback–Leibler divergence, on the other hand, measures the difference between the probability distributions of two datasets. Although a low value on such a test means higher statistical resemblance, in our experiment the score is taken as 1 minus the score of the Continuous Kullback–Leibler Divergence for consistency with the other scores.

Detection Metrics. The detection metrics measure how difficult it is to differentiate the synthetic data from the real one with a Machine Learning model. The real and the synthetic data are shuffled together and labeled with flags. Then the Machine Learning model is cross-validated by guessing if the data is real or not. Two Machine Learning models have been used – a support vector classifier and a logistic regression. The result is calculated as 1 minus the average ROC AUC score.

4 Results

In this section, we present the results of the experiment and outline the differences between the outputs of the three generators.

4.1 Statistical Significance of Differences

The distributions of the synthetic data, which are based on Dataset 2 have been plotted in **Fig. 3.** Example of different distributions of a column from Dataset 2 While Gretel.ai and Mostly AI follow a similar structure to the original dataset, SDV highly deviates from it.

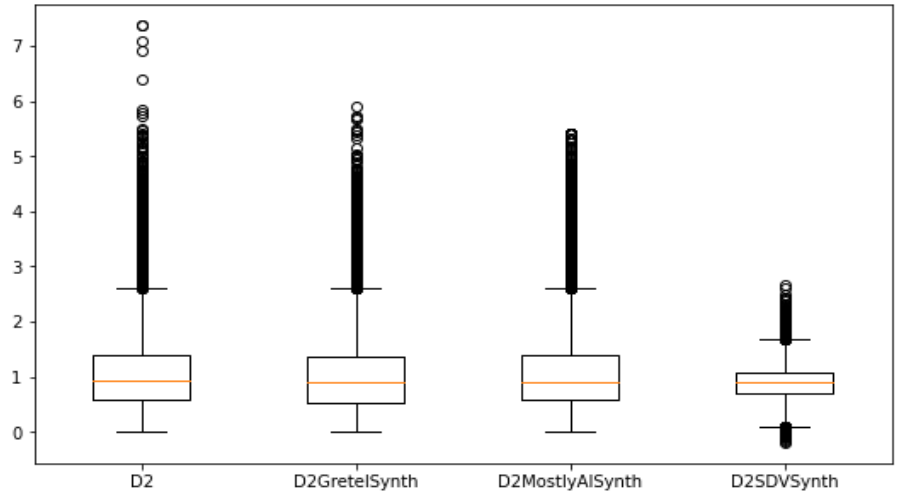

Fig. 3. Example of different distributions of a column from Dataset 2

The ANOVA test results can be seen in **Table 1.** Dataset 1-based ANOVA test results and **Table 2**. We find a statistically significant difference between the synthetic datasets for both datasets. The p-value of the data that originate from Dataset 1 is 0.000014 while the Dataset 2-based data scores 0.009653. Both values are lower than the predefined significance level of 0.05, which means that the hypothesis can be rejected. In addition, a post hoc test is conducted to further analyze the differences between datasets in a pairwise t-test. The p-values have been found to be lowest in the cases where the SDV synthetic dataset is involved.

Table 1. Dataset 1-based ANOVA test results

	sum_sq	**df**	**F**	**PR(>F)**
Group	2.756213e+07	2.0	73464.394638	0.000014
Residual	3.751767e+02	2.0	NaN	NaN

Table 2. Dataset 2-based ANOVA test results

	sum_sq	**df**	**F**	**PR(>F)**
Group	4.023541	2.0	4.713701	0.009653
Residual	126.757276	297.0	NaN	NaN

4.2 **Statistical Resemblance**

Table 3 gives an overview of the results of the Kolmogorov-Smirnov test. On Dataset 1 Mostly AI outperforms the other two generators with scores close to the sample results. Nevertheless, Gretel shows a high result as well – 0.938, while SDV has the lowest score of 0.804, which is far from the "satisfactory" range. Furthermore, Mostly AI performs equally well on the second dataset. The Gretel-generated data still reaches nearly 0.9 on that dataset and falls within range of the predefined tolerance. Finally, SDV-generated data have significantly worse performance on the KS test with only 0.643.

Table 3. Results of the Kolmogorov-Smirnov test

Method	Dataset 1	Dataset 2
Sample	0.985	0.992
Gretel	0.938	0.877
Mostly AI	0.984	0.987
SDV	0.804	0.643

The results from **Table 4** are similar to those from **Table 1** and represent the scores of the three generators on the continuous Kullback–Leibler divergence metric. Again, Mostly AI has the highest score with distribution similarity close to both of the samples. Gretel also appears to produce data, which imitates the distribution of its original counterpart with a score of 0.94. By contrast, SDV-generated data achieves only 68% distribution similarity.

Table 4. Results of the Continuous Kullback-Liebler Divergence test

Method	Dataset 1	Dataset 2
Sample	0.990	0.988
Gretel	0.940	0.944
Mostly AI	0.986	0.984
SDV	0.819	0.685

The scores on statistical metrics can be further explained by observing the different distributions in **Fig. 3.** Example of different distributions of a column from Dataset 2and the statistical properties in **Table 5**, which belong to a column from the second dataset and its synthetic equivalents. While Mostly AI and Gretel approximate the

mean and the standard deviation of the original dataset, SDV fails to do so successfully. Moreover, the generator allows values that are lower than 0, which cannot be found in its original counterpart.

Table 5. Mean and standard deviation example

Dataset	Dataset 2	Mostly AI	Gretel	SDV
Mean	1.070	1.062	1.014	0.910
Standard Deviation	0.695	0.705	0.659	0.318

4.3 Detected Resemblance

The results of the support vector classifier's detection have been summarized in **Table 6**. Testing the two samples against their respective datasets was impossible, as the tool crashes when it is set to run. Therefore, we define satisfactory results as those that are above 0.9. The tolerance is higher than the one for the statistical results for two main reasons. First, we cannot define a baseline value in the same way it is done for the statistical metrics. Second, the synthetic datasets have been generated with only basic parameter configuration of their respective generators, which might limit their capabilities. Therefore, setting a lower tolerance value seems unreasonable. It appears that Mostly AI is the most successful generator in deceiving the Machine Learning algorithm with scores close to 0.95 on both datasets. Gretel produces lower results with around 89% success on the first dataset and 92% on the second one. The score of the SDV-generated data, on the other hand, is notably lower than the other two generators – 0.63 on the first dataset and 0.53 on the second one.

Table 6. Results of the Support Vector Classifier Detection test

Method	Dataset 1	Dataset 2
Gretel	0.887	0.921
Mostly AI	0.945	0.944
SDV	0.636	0.530

Table 7 illustrates the performance of the linear regression model in distinguishing synthetic data from real one. Mostly AI's scores are similar to the other detection test. In the case of Gretel and SDV, the results are only slightly different than those from the SVC detection

Table 7. Results of the Logistic Detection test

Method	Dataset 1	Dataset 2
Gretel	0.850	0.881
Mostly AI	0.918	0.932
SDV	0.646	0.509

To further investigate the difference between the results of the three generators, the inter-field correlations have been explored. The correlation matrices, based on Dataset 1 can be seen on **Fig. 4**, **Fig. 5**, **Fig. 6** and **Fig. 7** (See Appendix). Mostly AI imitates the relationships between columns of the original dataset almost perfectly. Gretel, on the other hand, manages to capture the overall structure of the correlations only partially. Finally, the correlations of the SDV-based synthetic dataset are completely different than those found in the real dataset.

5 Discussion

The goal of our experiment is to measure and compare the utilities of the three synthetic data generators, namely Mostly AI, Gretel, and SDV, which are set to produce IoT data. The results show that out of the three generators Mostly AI achieves the highest scores on all metrics. Two conclusions can be drawn from this – first, the platform-based generator produces synthetic data while effectively preserving the statistical properties of the original dataset. Second, the data that is being generated imitates the original well, as the applied Machine Learning models failed to distinguish the synthetic data from the real one to a large extent. Therefore, Mostly AI proves to be efficient for generating realistic synthetic IoT data. A drawback of this generator is that the training process is time-consuming, as it took around 11 hours for each of the two datasets of the experiment.

Furthermore, the experiment shows that Gretel-generated data also emulates real data well in terms of its statistics. However, the performance on the resemblance metrics demonstrates that distinguishing the synthetic data from the real one is not as challenging for the Machine Learning models as doing so with Mostly AI synthetic data. As stated in [10] the utility of synthetic data is determined by its application. Therefore, if the Gretel generator is applied for a use case in which preserving the statistical properties is a priority, such as exploratory data analysis, then the utility of the Gretel generator would be satisfactory. Nevertheless, it is important to mention that the generators have been set with the recommended tuning of the parameters, which might have an effect on their performance.

Finally, the SDV generator is the least reliable generator out of the three. The statistical similarity that it produces is significantly lower than the other two generators

and the samples of the original dataset. Accordingly, the Machine Learning models are successful in distinguishing the synthetic values from the real ones. A possible explanation for this is that the generator fails to preserve the structure of the relationships in the original dataset. After further analyzing the distributions of the synthetic data, it is evident that SDV produces values that are significantly bigger or smaller and out of the range of values of the original datasets. As a consequence, the statistical properties of the synthetic data have been altered. In addition, the task to distinguish between real and synthetic values becomes easier, as the fake values are far from realistic. This leads to the conclusion that the SDV generator lacks the mechanisms to outline range constraints, thus the data that it synthesizes is not realistic. By contrast, both Mostly AI and Gretel remove outliers prior to testing. The difference in the results could be also attributed to the fact that the PAR model is tested with limited preprocessing. The model expects specific labeling of the columns depending on the role that they have in the dataset. Although the model can generate data by labeling only the time series column, the lower scores lead to the conclusion that the amount of contextual information is not enough to produce synthetic IoT data alone. Therefore, the SDV generator is deemed not suitable to reproduce IoT data.

5.1 **Limitations**

One of the challenges of this study was the scarce information about the SDV generator. Although there are available instructions on how to synthesize data, in-depth technical documentation is yet to be published. Having more knowledge about the architecture of the generator would give more insight into the results of this experiment. In addition, the PAR model is under active development, which means that future versions might be more successful in synthesizing IoT data.

Furthermore, in our study we explore only three generators, as time constraints limit the depth of this research. Many other similar generators could have been included. For example, GenRocket [18] and Hazy [19] are online tools, which satisfy the scope of this research, as they are not specifically intended for IoT data generation.

In addition, this study evaluates the synthetic data generators, by applying only two types of utility metrics. However, as already stated, utility is multidimensional and there are many other ways to measure it. For instance, other metrics from those described by [10], such as Hellinger distance or prediction accuracy would give another perspective on the synthetic data utility.

5.2 **Future Research**

The limitations of this paper create at the same time opportunities for future research. Firstly, as mentioned earlier, the parameters of the three generators have been tuned with only standard configuration. This experiment can be continued by tuning the parameters differently and exploring how different settings influence the results. In addition, in-depth documentation of the PAR model can help adjust it to produce

synthetic data with higher utility. Nevertheless, the model can still be explored by applying different levels of contextual information by labeling the columns accordingly.

Secondly, we recommend testing the Machine Learning efficacy of the generators. As stated earlier synthetic data comes as a solution for the lack of accessible IoT datasets with good quality and size. Therefore, future research could apply the methodology from [11] for comparing synthetic data on Mostly AI, Gretel, and SDV.

Thirdly, alternative metrics can be used to measure synthetic data utility. One approach would be to investigate the privacy levels of the synthetic data. This can be done by fitting an adversarial model on the generated data, which can predict predefined sensitive data points. Furthermore, the metrics described by [10], which have not been used in this paper can also be applied for testing.

Finally, the scope of the future research should be broadened beyond the three generators presented in this study. One of the contributions of our work is that it investigates the performance of said generators specifically within the IoT domain. Although most research done so far does not focus on IoT data, other established solutions to generate synthetic data can be explored and tested on IoT datasets.

6 Conclusion

Overall, the research question cannot be answered positively for all synthetic data generators. Mostly AI is found to be the best performing tool out of the three tested. The results shown by the generator are satisfactory for imitating real data and emulating statistical properties, according to our definition of it. Gretel, on the other hand, could be a suitable solution if it is intended to replicate statistics of real IoT data, but its output can be distinguished easier than by using Mostly AI. Finally, SDV does not yield high results on any of the utility metrics that have been explored, thus it can be deemed not fit for producing synthetic IoT data. We believe that this study can help researchers decide which tool to use for synthetic IoT data generation, thus contributing to the intensive research endeavors within the quickly gaining momentum IoT domain.

7 References

1. Balaji, S., Nathani, K., Santhakumar, R.: IOT technology, applications and challenges: A contemporary survey. Wireless Personal Communications. 108, 363–388 (2019).
2. Pradhan, B., Bhattacharyya, S., Pal, K.: IOT-based applications in healthcare devices. Journal of Healthcare Engineering. 2021, 1–18 (2021).
3. Shah, S.T., Badshah, F., Dad, F., Amin, N., Jan, M.A.: Cloud-assisted IOT-based smart respiratory monitoring system for asthma patients. Applications of Intelligent Technologies in Healthcare. 77–86 (2018).

4. Reddy, R.R., Mamatha, C., Reddy, R.G.: A review on machine learning trends, application and challenges in internet of things. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). (2018).
5. Emam, E.K., Mosquera, L., Hoptroff, R.: Practical Synthetic Data Generation: Balancing Privacy and the broad availability of data. O'Reilly, Beijing (2020).
6. Arlitt, M., Marwah, M., Bellala, G., Shah, A., Healey, J., Vandiver, B.: Iotabench. Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering. (2015).
7. Zhang, C., Kuppannagari, S.R., Kannan, R., Prasanna, V.K.: Generative Adversarial Network for synthetic time series data generation in smart grids. 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). (2018).
8. Dahmen, J., Cook, D.: SynSys: A Synthetic Data Generation System for Healthcare Applications. Sensors. 19, 1181 (2019).
9. Huang, Z., van Harmelen, F., ten Teije, A., Dentler, K.: Knowledge-based patient data generation. Lecture Notes in Computer Science. 83–96 (2013).
10. James, S., Harbron, C., Branson, J., Sundler, M.: Synthetic Data use: Exploring use cases to optimise data utility. Discover Artificial Intelligence. 1, (2021).
11. Hittmeir, M., Ekelhart, A., Mayer, R.: Utility and privacy assessments of synthetic data for regression tasks. 2019 IEEE International Conference on Big Data (Big Data). (2019).
12. Tomi. (2022, April 25). *Mostly AI, the Synthetic Data Company*. MOSTLY AI. Retrieved July 9, 2022, from https://mostly.ai/
13. The developer Stack for Synthetic Data, https://gretel.ai/, last accessed 2022/6/22.
14. Patki, N., Wedge, R., Veeramachaneni, K.: The synthetic data vault. 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). (2016).
15. A platform for Open Data of the European power system., https://open-power-system-data.org/, last accessed 2022/6/2.
16. Anttal, T.S.: Smart home dataset with weather information, https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information, last accessed 2022/6/13.
17. Seabold, S., & Perktold, J.: statsmodels: Econometric and statistical modeling with python. In 9th Python in Science Conference. (2010)
18. Self Service Synthetic Test Data, https://www.genrocket.com/, last accessed 2022/6/25.
19. Product, https://hazy.com/product/, last accessed 2022/6/25.

8 Appendix

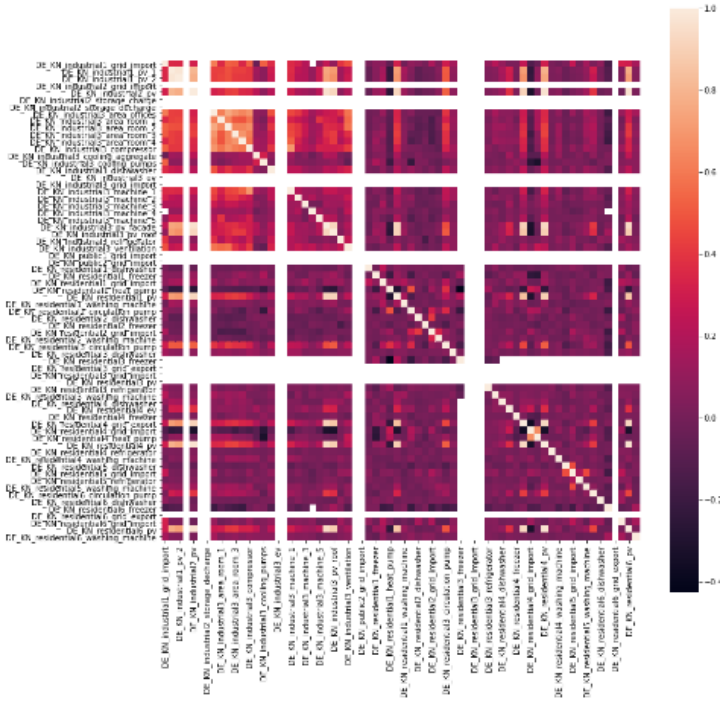

Fig. 4. Heatmap of Dataset 1

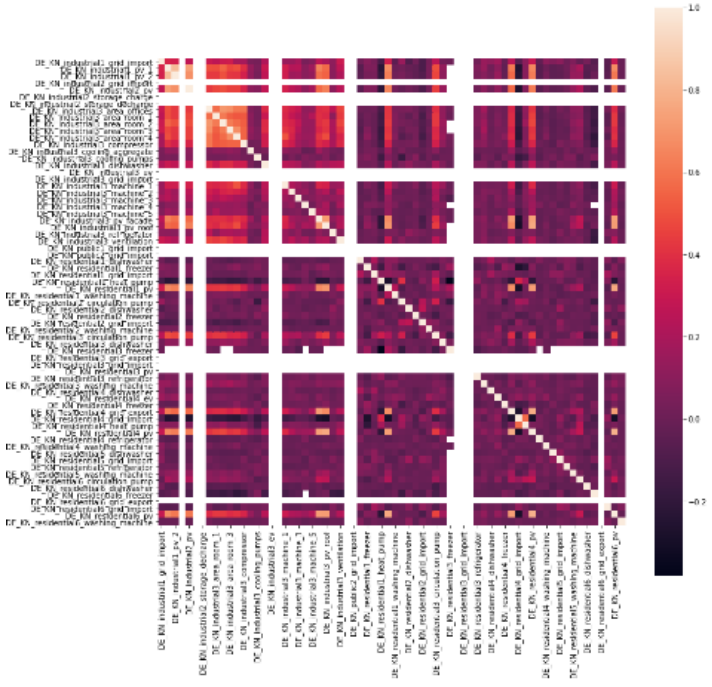

Fig. 5. Heatmap of Mostly AI-generated dataset

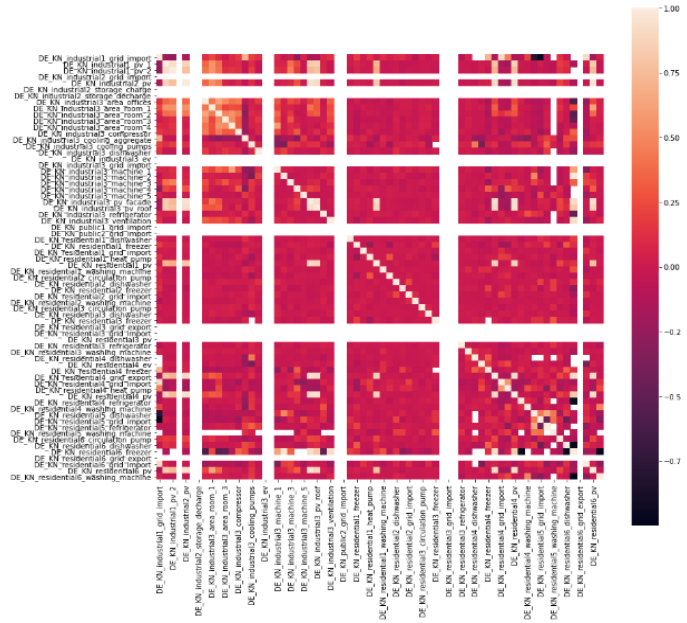

Fig. 6. Heatmap of Gretel-generated dataset

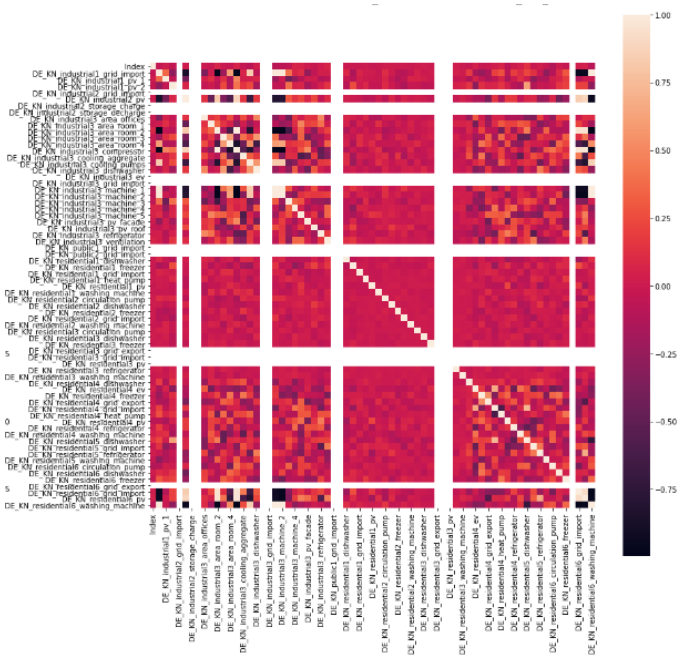

Fig. 7. Heatmap of SDV-generated dataset