Software Impacts 15 (2023) 100468

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Smart Meter Synthetic Data Generator development in python using FBProphet

Ezhilarasi P. [a,*], Ramesh L. [b], Xiufeng Liu [c], Jens Bo Holm-Nielsen [d]

[a] *Electronics & Communication Engineering, Dr. M.G.R. Educational and Research Institute, Chennai 95, India*
[b] *Electrical and Electronics Engineering, Dr. M.G.R. Educational and Research Institute, Chennai 95, India*
[c] *Department of Technology, Technical University of Denmark, Management and Economics, Lyngby, Denmark*
[d] *Department of Energy Technology, Aalborg University, 6700 Aalborg, Denmark*

ARTICLE INFO

Keywords:
Smart meter
Data generator
Time-series
Synthetic data

ABSTRACT

Data-science is a key component of modern science since it fuels AI, ML and data analytics, etc. As the electrical grid has been modernized into a smart grid, it has also become increasingly dependent on data science to monitor and control grid activity. Realistic data is essential to evaluating the algorithm's workability but it is difficult to obtain real smart meter data due to strict privacy and security policies of many countries. In this paper, using the prophet library, we code and develop a prediction-based Synthetic Data Generator GUI, which generate the synthetic data sets.

Code metadata

Current code version	*V1.1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-132
Permanent link to reproducible capsule	https://codeocean.com/capsule/0796106/tree/v1
Legal code license	*GNU General Public License (GPL)*
Code versioning system used	*git*
Software code languages, tools and services used	*Python*
Compilation requirements, operating environments and dependencies	*Python 3.9.12* *Pandas* *Tkinter* *Prophet*
If available, link to developer documentation/manual	
Support email for questions	ezhilarasihg@gmail.com

1. Introduction

Smart grid becomes the most researched area in the energy sector due to its tremendous benefits for consumers and utilities [1]. Smart grids are becoming more and more data-intensive as a result of the massive amounts of data (Big data) collected from all connected smart nodes in real time. Specifically, smart meters are essential to modernizing the electrical grid towards a smart grid. According to [2], big data will play a significant role in smart grids since more distributed energy systems and electric vehicles will be connected to the grid. With the growth of big data of issues, it is becoming increasingly necessary to have powerful algorithms to deal with them. As a consequence, smart meter data analytics studies were conducted in greater depth. In the field of smart meter data analytics, algorithms are tested using smart meter data in order to improve performance. To test the effectiveness of developed algorithms, a large amount of real-time data is required [3].

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.

E-mail addresses: ezhilarasi.restu@drmgrdu.ac.in (Ezhilarasi P.), ramesh.eee@drmgrdu.ac.in (Ramesh L.), xiuli@dtu.dk (X. Liu), jhn@energy.aau.dk (J.B. Holm-Nielsen).

https://doi.org/10.1016/j.simpa.2023.100468
Received 27 December 2022; Received in revised form 12 January 2023; Accepted 15 January 2023

Table 1

Paper	Analytic framework used	Processing scheme	Seasonality and trend removal
[5]	Spark	AR	Centered moving average
[2]	Spark	AR	Moving average
[6]	Generative Adversarial Network (GAN)	ARIMA	–

Unfortunately, many countries have privacy and security related policies that make it difficult to obtain smart meter data sets [4]. Synthetic data are widely utilized to mitigate the above problem without compromising privacy and security issues. The following Table 1. summarizes various research works related to synthetic data generation by various authors.

Nowadays, AI algorithms can be used to generate synthetic data, but they are randomly generated, and their likelihood of being similar to the original data is very low. There are many big data frameworks available for generating synthetic data, including Hadoop-Hbase, Cassandra, Elasticsearch, MongoDB, and Spark. In addition to that, a variety of time series analysis models have been used to produce the synthetic data above, including AR-Auto-Regressive, MA-Moving Average, ARMA-Auto-Regressive Moving Average, and ARIMA-Auto-Regressive Integrated Moving Average. In the above models, source data sets are used to predict the future data set based on time constraints. Stationarity conditions such as constant mean and variance should not be present in the source data set in order to achieve prediction for the input data set [7]. Moreover, the source data set should be free of seasonality and trends to obtain better predicted data set [6]. ARIM is the most commonly used model for synthetic data generation. However, it is important to select p, d, and q appropriately in preprocessing to remove seasonality and trends. Without this preprocessing, the predicted results will greatly differ from the original. Various programming languages are available for generating synthetic data generators, including Python, R, Scala, Julia, Java, SQL, MATLAB, and JavaScript. But out of all, Python is most predominantly used because it is open-source nature and its huge library support base, it is the most commonly used programming language. This paper models a Smart meter Synthetic Data Generator (Smart meter-SDG) using the Facebook prophet (FBprophet) library in Python. A main advantage of using Python is that it accepts pandas' libraries, which are ideal for handling large data sets. The Pandas framework is capable of tuning the data sets. Also, the two libraries are used differently based on the requirements of the program. In comparison to the above-mentioned library model to deal with prediction, the Prophet library offers the greatest advantage in that it automatically removes seasonality and trends. Prophet library eliminates the need for pre-calculation to remove trends, seasonality, and other factors that affect time series analysis. Using the FBprophet library, we code and develop a prediction-based Synthetic Data Generator GUI, which generate the synthetic data sets. The source CSV (real-time) file is used to generate synthetic data in CSV format depending upon the number meter and number days to be calculated. With FBprophet, time series data can be forecast based on an additive model that integrates seasonality, yearly, weekly, and daily trends, as well as holiday effects into non-linear trends. The algorithm is most effective when there are several seasons of historical data and strong seasonal effects in the data series.

2. Software description

2.1. Methodology

Programming for synthetic data generators involves two main parts: 1. Data cleaning 2. Synthetic Data generation. Tkinter, FBprophet, and Pandas are the main libraries used in this project. Time series prediction is performed using the FBprophet library. The data cleaning and processing of CSV files is done using Pandas and Tkinter is used to develop GUI for the proposed data generator. This source data set is obtained from UK Power Networks' project named "SmartMeter Energy Consumption Data in London Households" dated between November 2011 to February 2014, which was a 10 GB CSV file containing 167 million rows with 5600 unique half-hour energy consumption records [8]. Out of 5600 unique customer IDs, 20 consumer data sets are selected as input sources for synthetic data generation. The initial CSV file contains the user ID, tariff mode, as well as every half hour energy consumption data and datetime of the reading. Fig. 1 illustrate the working model of synthetic data generator.

2.2. Implementation

2.2.1. Pre-processing algorithm

As part of Pre-processing, the Hugh data source file is imported using the Pandas data Framework. Compared to Pandas, Pandas handles large data files efficiently without time delays. The initial part of the coding mainly focuses on separating the 20 different consumers based on consumer user ID. Each consumer's weekly energy consumption data is filtered and converted into a unique source CSV file. A similar procedure is followed for the remaining 20 consumers, resulting in 20 different source CSV files. These filtered files are used as input files for Smart meter-SDG and with that source file FBProphet library predict the results

Stage 1*. Importing raw source file with Pandas framework*

Stage 2*. Select 20 different consumers based on the consumer id and save it as separate CSV file using pandas frame work*

Stage 3*. Filter weekly data set from each consumer ID using filter command of pandas and store it as the source data file for data generator prediction*

Stage 4*. Repeat the stage 3 process for all 20 consumers to generate 20 different CSV source file*

2.2.2. Data generation

In this code block, the data from CSV will be processed to obtain the predicted results for the synthetic data generator. In order to interact with the user, the Tkinter library and its dependencies are incorporated to formulate the GUI. Two entry widgets are created to get data from the user for the number of meters and the number of days to be predicted. Based on the user entry of number of meters i.e n< = 20 out of 20 csv files n files are selected and feed as input file to the generator for prediction. The prediction is made through the FBPprophet library based on input of the number of days. It is not necessary to check the seasonality and trend in the data set since the library itself automatically removes them when working with the FBProphet library. CSV file should contain two columns named ds (data and time) and y (variable to be predicted using FBProphet).

As well, the ds column is converted to the actual date time data type before proceeding. Since the source data set is 30 minute-based, prediction is performed for a period of T = 48*t, where t = number of days. In total, 337 rows of energy consumption information are included in the input data set. A total of 337 data points are divided into 300 test points and 37 training points. In Python, machine learning algorithms and neural algorithms use trains and tests to predict future values. By using a pandas data frame, a number of separate csv files are created depending on the number of days. In addition to that, n number of CSV files also generated which contains all components predicted by the library.

Stage 1*. GUI is activated using Tkinter library*

Stage 2*. Users are asked with two input parameters to decide how many days and how many meters data needed*

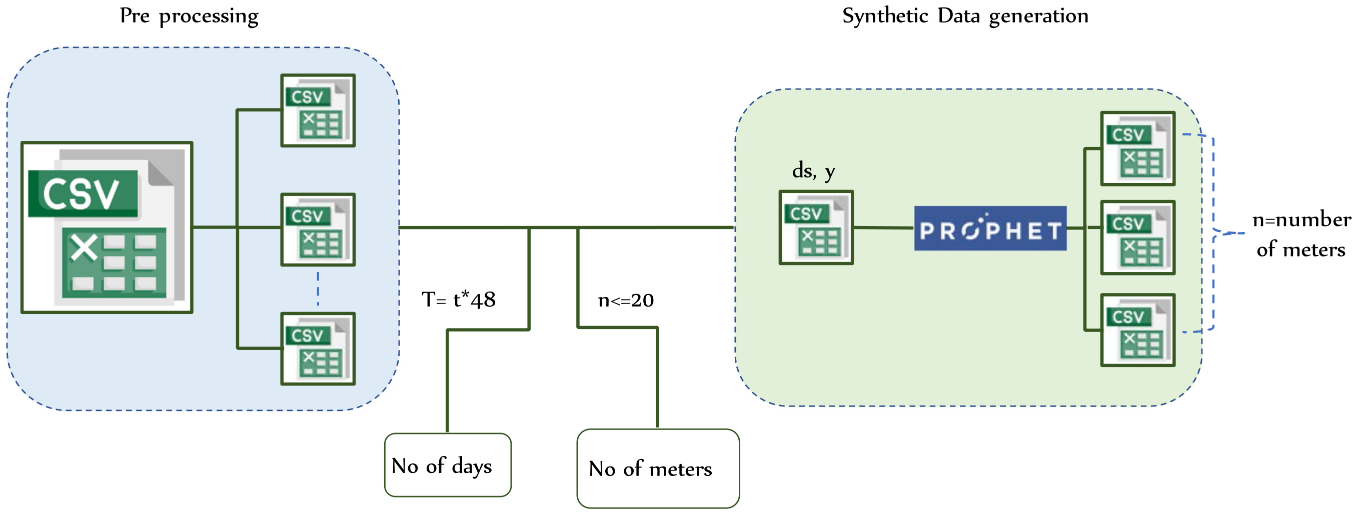
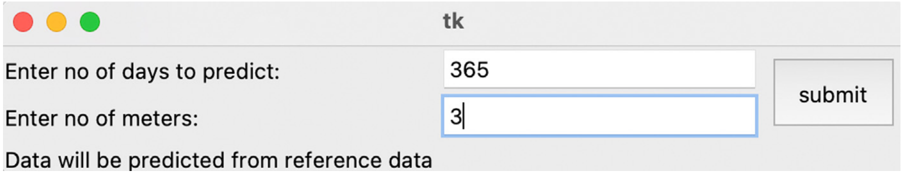


Fig. 1. Working model of data generator software.


tk

Enter no of days to predict: 365

Enter no of meters: 3

Data will be predicted from reference data

Fig. 2. GUI to get inputs from user.

Table 2

Papers	Application
[11]	Generating synthetic load patterns
[12]	Clustering approach
[13]	Synthetic data generation in smart home
[14]	Consumer privacy mitigation
[15]	Non-intrusive load monitoring in households
[16]	Demand Data Generation for ML
[17]	Synthetic Electric Power Systems
[18]	Smart meter data analytics

Stage 3 *After submission depending upon the no of meter input given by the user the number of CSV file to be loaded to the data generator is identified*

Stage 4 *From the source file data sets train and test data points are calculated to make the prediction*

Stage 5 *Depending upon the no of days (T), the prediction data set is generated using FBProphet library*

Stage 6 *Finally based on the no of meters (n), predicted results are saves as n different data set in the form of CSV file using panda's data frame*

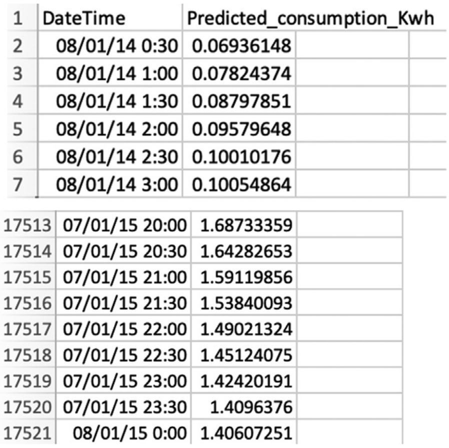
3. Impact overview

3.1. Real time applications

Multiple studies [9,10] have shown how synthetic data can help in smart grid data analytics to enhance the efficiency of scientific research and simulations. The research articles which highlight the necessity for synthetic data in smart grid data analytics for improvising the data analytic algorithms in smart grid are listed in the following Table 2.

Such a necessity was extended to demand side management with forecasted energy consumption to alert the consumer to shit their load to off peak. The energy consumption details are also forecasted using the synthetic data. The direct beneficiaries of the software we present in this study are researchers in smart grid data analytics and utilities who need to implement demand side management with forecasted energy consumption. The penetration of ML and AI in smart grid, researchers need to test various algorithms on smart grid model and management program. Developing successful AI and ML models requires access to large amounts of high-quality data. Hence, for their decision making process they are heavily dependent on real time data set to train the developed AI and ML. But the lack of such real time data had limited the range of research on smart grid analytics. To mitigate these problems there is a need of synthetic data which resembles the real time data closely. It was reported in a study [19] that 70% of the time, the synthetic data produced results that were on par with the real data.

With the developed software researcher can get the required amount of predicted or forecasted data set by interacting with the GUI with proper inputs. From https://github.com/pogog/synthetic-data-generator.git, the software framework can be download and executed with any python editing environment. After successful running of the framework desired output are generated in the form of CSV files for research purposes. Furthermore, our method can be easily adapted to demand side management with forecasted input (synthetic data) to improve the energy efficiency on the consumer side. DSM based on the historical data is one approach more commonly used in smart grid. But collecting and accessing that historical is very difficult and time consuming. In that scenario the develop software can forecast the data based on the real time data and it can be used for DSM. The developed Smart meter SDG uses FBProphet library to process very basic data set of energy consumption details without delay for T number days. In this software a simple GUI is developed to interact with user to get the necessary details from the user to run the program. Fig. 2 illustrate the GUI of developed software.



1	DateTime	Predicted_consumption_Kwh
2	08/01/14 0:30	0.06936148
3	08/01/14 1:00	0.07824374
4	08/01/14 1:30	0.08797851
5	08/01/14 2:00	0.09579648
6	08/01/14 2:30	0.10010176
7	08/01/14 3:00	0.10054864
17513	07/01/15 20:00	1.68733359
17514	07/01/15 20:30	1.64282653
17515	07/01/15 21:00	1.59119856
17516	07/01/15 21:30	1.53840093
17517	07/01/15 22:00	1.49021324
17518	07/01/15 22:30	1.45124075
17519	07/01/15 23:00	1.42420191
17520	07/01/15 23:30	1.4096376
17521	08/01/15 0:00	1.40607251

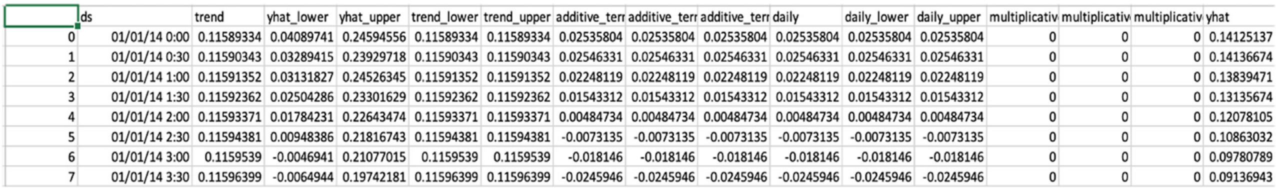
Fig. 3. Generated CSV data set with predicted value for 365 days.

3.2. Outcomes

Generally collecting real time data for testing is tedious and time consuming. For that synthetic data is the better option to testy the various algorithms in data analytics since it is readily available and can be collected without time delay. Furthermore, our software can handle a huge volume data set with less delay and produces the desired output CSV files. When compared with other methods stated in [2,5,20] in this developed software seasonality and trends are automatically removed from the time series source data set which results more simplest algorithm development with less number of coding to design the synthetic data generator. Fig. 3 shows that the CSV files for 3 houses are generated successfully for 365 days and Fig. 4 shows the all components predicted with FBProphet library.

In this developed software, synthetic data sets that are closely related to the original data set are predicted and generated based on the user's requirements and their comparative analysis is visualized in Fig. 5.

FBProphet library is used to generate the synthetic data set in which the challenges in time series data computations like seasonality and trends are automatically removed. This library reduces the computation



	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terr	additive_terr	additive_terr	daily	daily_lower	daily_upper	multiplicativ	multiplicativ	multiplicativ	yhat
0	01/01/14 0:00	0.11589334	0.04089741	0.24594556	0.11589334	0.11589334	0.02535804	0.02535804	0.02535804	0.02535804	0.02535804	0.02535804	0	0	0	0.14125137
1	01/01/14 0:30	0.11590343	0.03289415	0.23929718	0.11590343	0.11590343	0.02546331	0.02546331	0.02546331	0.02546331	0.02546331	0.02546331	0	0	0	0.14136674
2	01/01/14 1:00	0.11591352	0.03131827	0.24526345	0.11591352	0.11591352	0.02248119	0.02248119	0.02248119	0.02248119	0.02248119	0.02248119	0	0	0	0.13839471
3	01/01/14 1:30	0.11592362	0.02504286	0.23301629	0.11592362	0.11592362	0.01543312	0.01543312	0.01543312	0.01543312	0.01543312	0.01543312	0	0	0	0.13135674
4	01/01/14 2:00	0.11593371	0.01784231	0.22643474	0.11593371	0.11593371	0.00484734	0.00484734	0.00484734	0.00484734	0.00484734	0.00484734	0	0	0	0.12078105
5	01/01/14 2:30	0.11594381	0.00948386	0.21816743	0.11594381	0.11594381	-0.0073135	-0.0073135	-0.0073135	-0.0073135	-0.0073135	-0.0073135	0	0	0	0.10863032
6	01/01/14 3:00	0.1159539	-0.0046941	0.21077015	0.1159539	0.1159539	-0.018146	-0.018146	-0.018146	-0.018146	-0.018146	-0.018146	0	0	0	0.09780789
7	01/01/14 3:30	0.11596399	-0.0064944	0.19742181	0.11596399	0.11596399	-0.0245946	-0.0245946	-0.0245946	-0.0245946	-0.0245946	-0.0245946	0	0	0	0.09136943

Fig. 4. Generated CSV data set with all components of predicted value for 365 days.

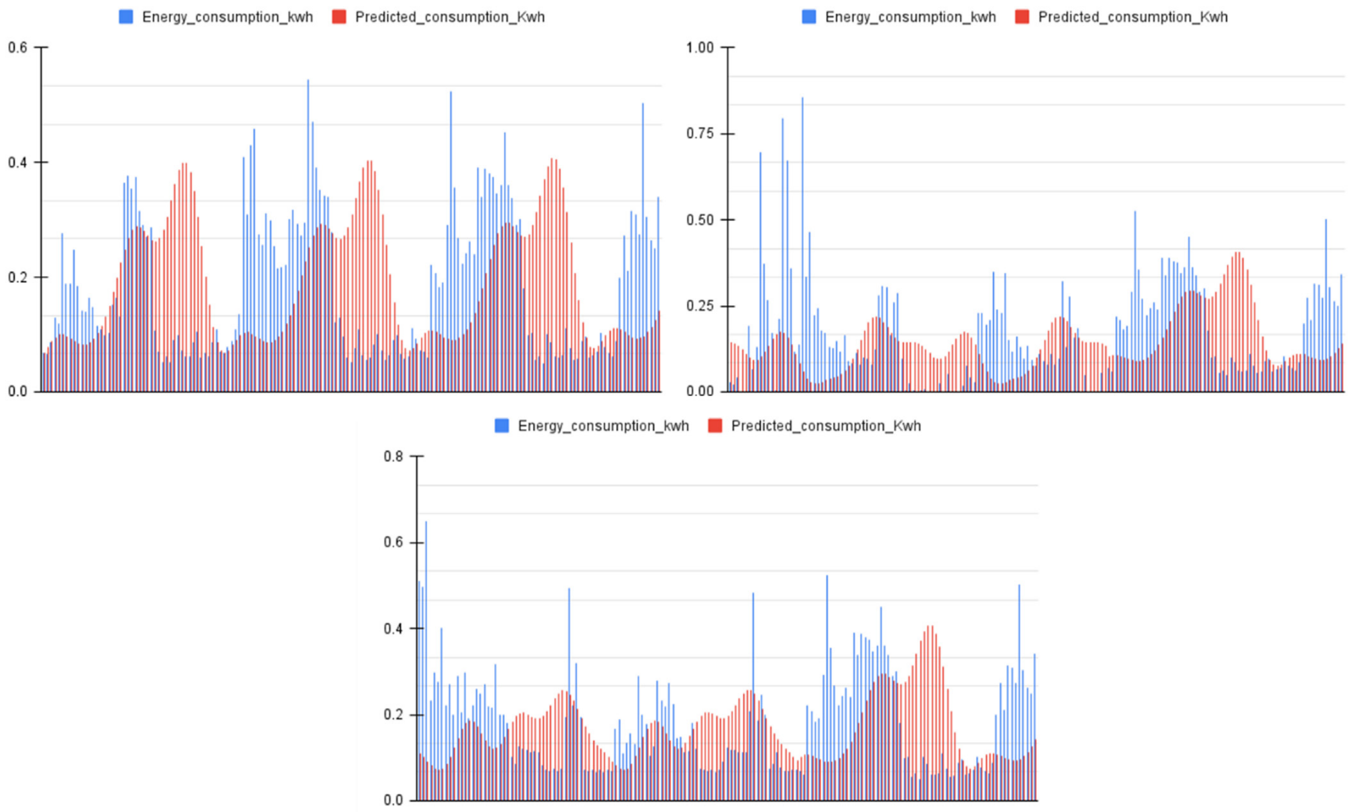


Fig. 5. Comparison of actual consumption with predicted data in different generated CSV files.

time and complexity in the developed software to produce the synthetic data. Scientists, researchers, and developers can test smart meter algorithms using synthetic data sets without compromising privacy or security. This software can be further developed with some more controlling parameter like seasonal selection, cluster selection etc.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Jain, K.N. Vinoth, A. Paventhan, V. Kumar Chinnaiyan, V. Arnachalam, M. Pradish, Survey on smart grid technologies-smart metering, IoT and EMS, in: 2014 IEEE Students' Conf. Electr. Electron. Comput. Sci. SCEECS 2014, 2014.

[2] M.H. Ansari, V. Tabatab Vakili, B. Bahrak, Evaluation of big data frameworks for analysis of smart grids, J. Big Data 6 (1) (2019).

[3] T. Sirojan, S. Lu, B.T. Phung, E. Ambikairajah, Embedded edge computing for real-time smart meter data analytics, in: 2019 International Conference on Smart Energy Systems and Technologies, SEST, 2019, pp. 1–5.

[4] D. Lee, D.J. Hess, Data privacy and residential smart meters: Comparative analysis and harmonization potential, Util. Policy 70 (2021) 101188.

[5] N. Iftikhar, X. Liu, S. Danalachi, F.E. Nordbjerg, J.H. Vollesen, A scalable smart meter data generator using spark, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 10573 LNCS, 2017, pp. 21–36.

[6] C. Zhang, S.R. Kuppannagari, R. Kannan, V.K. Prasanna, Generative adversarial network for synthetic time series data generation in smart grids, in: 2018 IEEE Int. Conf. Commun. Control. Comput. Technol. Smart Grids, SmartGridComm 2018, 2018, pp. 1–6.

[7] S. Mohanasundaram, G.S. Kumar, B. Narasimhan, A novel deseasonalized time series model with an improved seasonal estimate for groundwater level predictions, H2Open J. 2 (1) (2019) 25–44.

[8] U. power Network, Smartmeter energy consumption data in London households, 2015, [Online]. Available: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households.

[9] C. Zhang, S.R. Kuppannagari, R. Kannan, V.K. Prasanna, Generative adversarial network for synthetic time series data generation in smart grids, in: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm, 2018.

[10] S. Asre, A. Anwar, Synthetic energy data generation using time variant generative Adversarial Network, Electronics 11 (3) (2022) 355.

[11] S.E. Kababji, P. Srikantha, A data-driven approach for generating synthetic load patterns and usage habits, IEEE Trans. Smart Grid 11 (6) (2020) 4984–4995.

[12] K. Mason, S. Vejdan, S. Grijalva, An 'on the fly' framework for efficiently generating synthetic big data sets, in: 2019 IEEE International Conference on Big Data, Big Data, 2019.

[13] M. Razghandi, et al., Variational autoencoder generative adversarial network for Synthetic Data Generation in smart home, in: ICC 2022 - IEEE International Conference on Communications, 2022.

[14] S. Desai, et al., Mitigating consumer privacy breach in smart grid using obfuscation-based generative Adversarial Network, Math. Biosci. Eng. 19 (4) (2022) 3350–3368.

[15] C. Klemenjak, et al., A synthetic energy dataset for non-intrusive load monitoring in households, Sci. Data 7 (1) (2020).

[16] B. Yilmaz, R. Korn, Synthetic demand data generation for individual electricity consumers: Generative Adversarial Networks (GANs), Energy AI 9 (2022) 100161.

[17] H. Li, et al., The creation and validation of load time series for synthetic electric power systems, IEEE Trans. Power Syst. 36 (2) (2021) 961–969.

[18] J.N. Kahlen, A. Muhlbeier, M. Andres, A. Moser, B. Rusek, D. Unger, K. Kleinekort, Synthetic Data – A Solution to Train Diagnostic Systems for High-Voltage Equipment without Fault-Condition Measurements, CIGRE Sci. Eng. 2022 (24) (2022) 1–28.

[19] N. Patki, R. Wedge, K. Veeramachaneni, The synthetic data vault, in: 2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA, 2016.

[20] N. Iftikhar, X. Liu, F.E. Nordbjerg, S. Danalachi, A prediction-based smart meter data generator, in: 2016 19th International Conference on Network-Based Information Systems, NBiS, 2016, pp. 173–180.