METHOD ARTICLE

Synthetic time series data generation for edge analytics

[version 1; peer review: awaiting peer review]

Subarmaniam Kannan

Faculty of Information Science and Technology, Multimedia University, Malacca, 75450, Malaysia

V1 First published: 20 Jan 2022, 11:67
<https://doi.org/10.12688/f1000research.72984.1>
Latest published: 20 Jan 2022, 11:67
<https://doi.org/10.12688/f1000research.72984.1>

Open Peer Review

Approval Status *AWAITING PEER REVIEW*

Any reports and responses or comments on the article can be found at the end of the article.

Abstract

Background: Internet of Things (IoT) edge analytics enables data computation and storage to be available adjacent to the source of data generation at the IoT system. This method improves sensor data handling and speeds up analysis, prediction, and action. Using machine learning for analytics and task offloading in edge servers could minimise latency and energy usage. However, one of the key challenges in using machine learning in edge analytics is to find a real-world dataset to implement a more representative predictive model. This challenge has undeniably slowed down the adoption of machine learning methods in IoT edge analytics. Thus, the generation of realistic synthetic datasets can leverage the need to speed up methodological use of machine learning in edge analytics.

Methods: We create synthetic data with features that are like data from IoT devices. We use an existing air quality dataset that includes temperature and gas sensor measurements. This real-time dataset includes component values for the Air Quality Index (AQI) and ppm concentrations for various polluting gases. We build a JavaScript Object Notation (JSON) model to capture the distribution of variables and the structure of this real dataset to generate the synthetic data. Based on the synthetic dataset and original dataset, we create a comparative predictive model.

Results: Analysis of synthetic dataset predictive model shows that it can be successfully used for edge analytics purposes, replacing real-world datasets. There is no significant difference between the real-world dataset compared the synthetic dataset. The generated synthetic data requires no modification to suit the edge computing requirements.

Conclusions: The framework can generate representative synthetic datasets based on JSON schema attributes. The accuracy, precision, and recall values for the real and synthetic datasets indicate that the logistic regression model is capable of successfully classifying data.

Keywords

Synthetic data generation, Internet of Things, edge analytics, predictive model, machine learning

This article is included in the **Artificial Intelligence and Machine Learning** gateway.

This article is included in the **Research Synergy Foundation** gateway.

Corresponding author: Subarmaniam Kannan (subar.kannan@mmu.edu.my)

Author roles: Kannan S: Methodology

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the Ministry of Higher Education (MOHE) Fundamental Research Grant Scheme (FRGS/1/2019/ICT03/MMU/03/2).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2022 Kannan S. This is an open access article distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Kannan S. **Synthetic time series data generation for edge analytics [version 1; peer review: awaiting peer review]** F1000Research 2022, **11**:67 <https://doi.org/10.12688/f1000research.72984.1>

First published: 20 Jan 2022, **11**:67 <https://doi.org/10.12688/f1000research.72984.1>

Introduction

The widespread adoption of the Internet of Things (IoT) in business and industry has resulted in significant investment in advanced applications development (Brous *et al.*, 2020). These applications focus on increasing efficiency and cost reduction while speeding up the analytic process at receiving ends. However, initial IoT data generation and dependence on cloud-based data storage and data processing have limited the success with IoT applications. “Roughly 10% of enterprise-generated data is processed outside of an established centralised data centre or cloud,” according to Gartner. By 2025, it is predicted that this number would have grown to 75% (Van der Meulen, 2018).

As a result, Edge Computing (EC) is emerging as a key enabler technology for network edge-based analytics and real-time decision-making. Edge computing places processing and analytics capability close to the source of the data. While extracting high-level data from the raw sensory input, this strategy reduces network latency. Integration of Machine Learning (ML) capabilities into EC has enabled sensor-based application specific analytics at the IoT network edge. Due to technological advancements in computer processor power, energy efficiency, memory capacity, and device size downsizing, machine learning computation can now be performed at edge nodes (Murshed *et al.*, 2022).

The development of ML-based edge analytics for IoT applications is unlike that of traditional machine learning due to the hardware limitations and lack of sensory data availability associated with ML-based edge analytics (Li *et al.*, 2021). Finding real-world datasets that reflect sensory data for a prediction model is one of the troublesome issues in ML-based edge analytics development (Chen *et al.*, 2020). This issue is undeniably hampering the rapid adoption of machine learning methods in IoT Edge analytic integration.

Thus, realistic synthetic dataset generation can leverage the need to speed up the methodological use of machine learning in edge analytics. Synthetic data is information that is artificially designed to represent real-world events (Nikolenko, 2019). Researchers have used various techniques such as stochastic process (Salim *et al.*, 2018), rule-based data generation (Jeske *et al.*, 2005) and deep generative model (Alzantot *et al.*, 2017) to generate synthetic data for many applications (Anderson *et al.*, 2014). Because of privacy and legality concerns, data science has given rise to synthetic data. Synthetic data has been widely used to supplement the ever-increasing demand for data science to predict various global business and environmental phenomena (Howe *et al.*, 2017). When a need for previously unheard-of real-world datasets arises, the importance of synthetic data increases to criticality. Yet, the use of synthetic datasets has become difficult when compounded by the need for privacy preservation. Due to a scarcity of genuine datasets, medical researchers have been employing synthetic data to evaluate medical applications in conditional and control environments (Azizi *et al.*, 2021). Similarly, self-driving vehicle re-identification technology uses randomized synthetic data as a precursor to training vehicles (Tang *et al.*, 2019).

The scarcity of getting accurate IoT generated datasets to represent localized real-world environments and the inaccuracy of pre-defined or public repository datasets has driven IoT researchers to generate synthetic data to cater to their unique requirements (Sengupta & Chinnasamy, 2020; Tazrin, 2021; Zuolkernan *et al.*, 2021).

There are several benefits that the use of synthetic data brings when compensating for unavailable real-world data. Firstly, the advent of personal and community privacy advocacy has limited many developers and researchers in using real datasets (Oh *et al.*, 2020). The creation of synthetic data that represents details of real-world data, including distributions, non-linear relationships, and noise, can eliminate legal issues (Tucker *et al.*, 2020). Instead of using costly real-world data to generate a predictive model, synthetic data are able to cater to various possible desired predictive events found in real datasets (Jordon *et al.*, 2018).

The remainder of the paper is structured as follows. Section 2 discusses related work in the area of synthetic data creation. Section 3 discusses the JSON-based synthetic data generation framework. Section 4 presents the generated synthetic datasets and the results of the synthetic data validation experiment. Section 5 outlines the conclusions.

Related work

According to Coimbra *et al.* (2020), the difficulty in training the machine learning model for detection and classification with suitable datasets that cover all the possibilities of a domain may be expensive and could be associated with privacy concerns. Thus, synthetic data offers a potential answer in terms of lowering the cost of data acquisition while also addressing data privacy concerns.

The bulk of synthetic data creation methods rely on extracting properties from existing real-world datasets. These extracted features from a real dataset are utilised to create a new dataset. Several criteria are used as guidelines to generate synthetic data. Deep Neural Network (DNN)-based methods include Auto-Encoders (AE) and Generative Adversarial Networks (GAN) (Frid-Adar *et al.*, 2018).

However, [Torres \(2018\)](#) point out that the suggested generation process's pattern identification phase states that after determining (quantitatively) the impact of each feature, the dataset columns are ordered according to how important they are. Priority may be assigned in the following order: most influential to least influential, or vice versa. This strategy may not be optimal or time-efficient, and so may have a detrimental influence on training and processing durations if a computationally expensive characteristic is checked at the start of the cue.

However, some researchers create synthetic data based using actual data formats such as Comma-Separated Values (CSV), Java Script Object Notation (JSON), and Extensible Markup Language (XML). [Anderson *et al.* \(2014\)](#) developed and built a Hadoop-based synthetic IoT data creation system capable of producing vast amounts of data. Using the Document Type Definition (DTD) and the recreation synthesis set, this system extracts data structures from IoT XML data. Through different XML data extensions, this modular system supports many statistical distributions.

JSON data types are human-readable data and may represent metadata and computer-readable data ([Sun *et al.*, 2020](#)). The JavaScript Language comprises pre-defined data structures, such as arrays that provide order lists and objects that represent name-value pairs. JSON metadata offer an effective method to examine the structure and variables of any data file, particularly a big dataset. In contrast to XML, which is a native format of the JavaScript language, JSON-based datasets are simple to read and represent a data exchange language. Many computer languages such as C++, C#, Cold Fusion, Java, Perl, PHP, Python and Ruby support it for online data integration.

The features of IoT data are critical in the creation and structure of synthetic data and can be classified based on the method through which the data is collected. Applications such as IoT-based digital healthcare systems ([Ed-daoudy & Maalmi, 2019](#)) and autonomous vehicle monitoring systems ([Kavitha & Ravikumar, 2020](#)), generate huge amounts of data in continuous streams. Additionally, it is important to consider how the massive data was gathered and stored away from IoT and edge devices. These massive databases are processed and analysed in order to forecast long-term business trends. Streaming datasets, on the other hand, require rapid pre-processing and analytics in order to extract immediate and relevant information for a speedy decision, like in the autonomous vehicle example ([Elsaleh *et al.*, 2020](#)). Time-series data is used to generate a large quantity of streaming IoT data ([Kumar *et al.*, 2020](#)). Data from streaming IoT devices is collected and analysed depending on the time it was generated. The ability to synchronise time across a pool of IoT devices spread over a large monitoring environment will be critical in evaluating the quality of this data ([Ferrari *et al.*, 2020](#)).

Methods

Synthetic data generation framework

The primary objective of this synthetic data creation is to provide an experimental framework that may train a machine learning prediction model. The framework can create correct JSON schema characteristics based on synthetic datasets. This section introduces the IoT synthetic data creation framework, which is divided into two stages. The first stage involves extracting structure and variables from the original time series dataset in comma-separated values (.CSV) format. In the second stage, the extracted structure and variables from the original datasets are used to develop and construct synthetic CSV datasets. [Figure 1](#) shows a schematic diagram that depicts the framework for generating synthetic data.

The original dataset for this experiment (Air Quality Data in India 2015-2020) was obtained from the [Kaggle](#) dataset repository (see *Underlying data*). We choose this dataset based on its data format (CSV) and because it is publicly available online. The dataset contains air quality data and AQI (Air Quality Index) for different stations in 26 Indian cities, taken on an hourly basis. The dataset contains different air quality indices that are rated in parts per million (ppm) as satisfactory (80-99), moderate (100-199), poor (200-299), and very poor (300-400). These categories reflect the quality of the air based on data collected from dataset and will be utilised as predictive components in the trained model.

The dataset is formatted in a CSV file with several columns that influence the type of categories that the air quality will be, such as "StationId" for the station where the sensors were placed, "PM2.5 and PM10" for the number of particles with 2.5 micrometres and 10 micrometres, and the rest are the concentration of chemicals and gases in the air that will influence the AQI. [Table 1](#) shows the data components of the AQI dataset.

The CSV file containing the original AQI dataset is converted to JSON using the freely available online CSV to JSON parser, [CSVJSON](#) ([Drapeau *et al.*, 2014](#)). After that, the output is saved as a text file (csvjson1.txt). The original dataset's JSON output contains arrays of data objects but no structure or syntax. To enable the structure and variables to be identified, the JSON file is converted to JSON Schema using the freely available [Online JSON to JSON Schema Converter](#) ([Liquid Technologies Limited, 2001](#)). The JSON schema output demonstrates the syntax and structure's

Figure 1. Synthetic data generation framework based on JavaScript Object Notation (JSON) data format. CSV, Comma-Separated Values.

Table 1. Data elements used to calculate Air Quality Index.

Dataset variables
City
Data
Particulate Matter 2.5-micrometer in ug/m ³
Particulate Matter 10-micrometer in ug/m ³
Nitric Oxide in ug/m ³
Nitric Dioxide in ug/m ³
Any Nitric x-oxide in ppb
Ammonia in ug/m ³
Carbon Monoxide in mg/m ³
Sulphur Dioxide in ug/m ³
Ozone or trioxygen
Benzene
Toluene
Xylene

accuracy and the file's entire structure and variable components. The JSON Schema in [Figure 2](#) is derived from the JSON file containing the original AQI dataset.

Once the variable components and structure of the AQI dataset JSON schema have been identified, they are mapped and written into the [Python Faker](#) (version 8.1.2) data generator application ([Faraglia, 2017](#)). The Faker data generator application generates 240000 synthetic data records in CSV format for this experiment. We limited the size to facilitate the extraction and processing of the output structure and variables.

The synthetic dataset's CSV file is then converted to JSON using the online CSV to JSON parser. After that, the output is saved as a text file (csvjson2.txt). The next step is to use the Online-JSON-to-JSON schema-converter to convert the JSON structured text file to JSON Schema.

The JSON schema for this synthetic dataset is verified by ensuring that the structure and variable elements match the JSON schema for the actual AQI dataset. This is the first step in the validation process, and it is used to determine the

<pre>{ "\$schema": "http://json-schema.org/draft-04/schema#", "type": "array", "items": [{ "type": "object", "properties": { "StationId": { "type": "string" }, "Datetime": { "type": "string" }, "PM2.5": { "type": "number" }, "PM10": { "type": "integer" }, "NO": { "type": "number" }, "NO2": { "type": "number" }, "NOx": { "type": "number" }, "NH3": { </pre>	<pre> "type": "number" }, "CO": { "type": "number" }, "SO2": { "type": "number" }, "O3": { "type": "number" }, "Benzene": { "type": "number" }, "Toluene": { "type": "number" }, "Xylene": { "type": "number" } }, "required": ["StationId", "Datetime", "PM2.5", "PM10", "NO", "NO2", "NOx", "NH3", </pre>	<pre> "CO", "SO2", "O3", "Benzene", "Toluene", "Xylene"] },], }</pre>
--	--	---

Figure 2. Parsed JavaScript Object Notation (JSON) schema from Air Quality Index (AQI) dataset. PM2.5, number of particles that are 2.5 micrometres; PM10, number of particles that are 10 micrometres; NO, nitric oxide; NO₂, nitric dioxide; NO_x, nitrogen oxides; NH₃, ammonia; CO, carbon monoxide; SO₂, sulphur dioxide; O₃, ozone.

quality of the generated synthetic dataset. The JSON Schema structure for the generated synthetic dataset is depicted in Figure 3; see *Underlying data* (Kannan, 2021a).

For this experiment, accuracy will be measured as the proportion of accurate predictions of the given air quality index. Precision will be determined by the proportion of accurate classifications. For example, how many of the air quality indicators in the 'poor' category were properly classified? To recall, it will be evaluating the model itself on how much of the data was properly classified given the air quality index. The ideal model would strive for high accuracy, precision, and recall value.

<pre>{ "\$schema": "http://json-schema.org/draft-04/schema#", "type": "array", "items": [{ "type": "object", "properties": { "No": { "type": "string" }, "StationId": { "type": "string" }, "Datetime": { "type": "string" }, "PM2.5": { "type": "number" }, "PM10": { "type": "number" }, "NO": { "type": "number" }, "NO2": { "type": "number" }, "NOx": { </pre>	<pre> "type": "number" }, "NH3": { "type": "number" }, "CO": { "type": "number" }, "SO2": { "type": "number" }, "O3": { "type": "number" }, "Benzene": { "type": "number" }, "Toluene": { "type": "number" }, "Xylene": { "type": "number" }, "AQI": { "type": "integer" } }, "required": ["No", "StationId", </pre>	<pre> "Datetime", "PM2.5", "PM10", "NO", "NO2", "NOx", "NH3", "CO", "SO2", "O3", "Benzene", "Toluene", "Xylene", "AQI"] },], }</pre>
--	---	--

Figure 3. The JavaScript Object Notation (JSON) Schema structure for the generated synthetic dataset. PM2.5, number of particles that are 2.5 micrometres; PM10, number of particles that are 10 micrometres; NO, nitric oxide; NO₂, nitric dioxide; NO_x, nitrogen oxides; NH₃, ammonia; CO, carbon monoxide; SO₂, sulphur dioxide; O₃, ozone; AQI, Air Quality Index.

Table 2. Linear regression model performance for real and synthetic Air Quality Index (AQI) datasets.

	Original dataset	Synthetic dataset
Accuracy (%)	98.18	99.21
Precision (%)	98.18	99.21
Recall (%)	98.18	99.21

Figure 4. The confusion matrix performance of the classification model for both the original Air Quality Index (AQI) and synthetic datasets.

Results

The full analysis code is available in *Software availability* (Kannan, 2021b). To open the files, use [Jupyter](#). Python Faker generated approximately 240000 synthetic data points for this experiment purpose. The first visible comparison is based on the JSON Schema for the original dataset and the JSON Schema for the synthetic dataset, both of which appear to be quite precise in matching the same structure and variable components. We further validated our experimental approach by training a machine learning model to predict the four air quality categories specified in the dataset using both the actual AQI and a synthetic dataset.

The [Python Scikit Learn](#) library (Pedregosa *et al.*, 2011) is used to train the both datasets using the Logistic Regression machine learning model. Logistic Regression appears to be the optimal machine learning model because it accounts for the possibility that the given condition or variable determines the air quality categories. For instance, if the air quality index (AQI) and other variables are between 80 and 99, the air quality is considered satisfactory.

The model's performance characteristics, such as accuracy, precision, and recall, vary according to the AQI dataset used. The effectiveness of the Linear Regression model in predicting both real and synthetic AQI datasets is demonstrated in [Table 2](#). The synthetic dataset appears to outperform the real dataset by 1.03 percent; this difference may be due to the nature of the dataset that we did not remove the missing data. The synthetic dataset performs slightly better than the original dataset because the data is nearly identical to the original, and the original dataset appears to contain an error in which some rows are incomplete. This incomplete data is due to sensor failure at one of the stations that caused partial recording. A confusion matrix is constructed to explain the classification model's performance on the two datasets. Nevertheless, as demonstrated by the confusion matrix, the model is effective at classification. The confusion matrix performance of the classification model is shown in [Figure 4](#) for both the original AQI and synthetic datasets.

Conclusions

In this paper, we present the synthetic data generation framework and its experimental results. Based on JSON schema attributes, the framework can produce correct synthetic datasets. We use both the original and synthetic datasets to train

the machine learning model, in addition to early schema validation. The logistic regression model appears to be capable of successfully handling the classification process, as the accuracy, precision, and recall scores for both the real and synthetic datasets were approximately 98 percent. A predictive comparison model is developed utilising synthetic and original datasets based on the benchmark dataset. Analysis of the synthetic dataset predictive model shows that it can be successfully deployed to edge analytics to replace real-world datasets. There is no significant difference between real-world datasets compared to the synthetic dataset.

Data availability

Underlying data

The Air Quality Data in India (2015-2020) dataset used in this study is freely available on Kaggle: <https://www.kaggle.com/rohanrao/air-quality-data-in-india>. Access requires free registration to Kaggle and agreement to the terms of use.

Zenodo: Synthetic time series data generation for edge analytics. <https://doi.org/10.5281/zenodo.5673924> (Kannan, 2021a).

This project contains the following underlying data:

- Json Schema for Original Datasets.docx (JSON Schema showing original sensors data variables).
- Json Schema for Synthetic Dataset.txt (JSON Schema showing generated synthetic data variables).

Data are available under the terms of the [Creative Commons Attribution 4.0 International license](#) (CC-BY 4.0).

Software availability

Source code available from: <https://github.com/Subar1/synthetic/tree/1.0>

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.5726027> (Kannan, 2021b)

License: MIT

References

- Alzantot M, Chakraborty S, Srivastava M: **Sensegen: A deep learning architecture for synthetic sensor data generation**. 2017 *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2017; 188–193.
- Anderson JW, Kennedy KE, Ngo LB, et al.: **Synthetic data generation for the internet of things**. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*. 2014. [Publisher Full Text](#)
- Azizi Z, Zheng C, Mosquera L, et al.: **Can synthetic data be a proxy for real clinical trial data? A validation study**. *BMJ Open*. 2021; **11**(4): e043497. [PubMed Abstract](#) | [Publisher Full Text](#)
- Brous P, Janssen M, Herder P: **The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations**. *International Journal of Information Management*. 2020; **51**: 101952. [Publisher Full Text](#)
- Chen L, Xu Y, Lu Z, et al.: **IoT Microservice Deployment in Edge-cloud Hybrid Environment Using Reinforcement Learning**. *IEEE Internet of Things Journal*. 2020. [Publisher Full Text](#)
- Coimbra A, Neto C, Ferreira D, et al.: **Review of Trends in Automatic Human Activity Recognition in Vehicle Based in Synthetic Data**. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 12490 LNCS. 2020. [Publisher Full Text](#)
- Ed-daoudy A, Maalmi K: **A new Internet of Things architecture for real-time prediction of various diseases using machine learning on big data environment**. *Journal of Big Data*. 2019; **6**(1): 104. [Publisher Full Text](#)
- Elsaleh T, Enshaeifar S, Rezvani R, et al.: **IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams and Its Use with Data Analytics and Event Detection Services**. *Sensors*. 2020; **20**(4): 953. [PubMed Abstract](#) | [Publisher Full Text](#)
- Faraglia DO. Contributors: *Faker [Computer software]*. 2017, October 17. [Reference Source](#)
- Ferrari P, Bellagente P, Depari A, et al.: **Evaluation of the impact on industrial applications of NTP Used by IoT devices**. 2020 *IEEE International Workshop on Metrology for Industry 4.0 IoT*. 2020; 223–228. [Publisher Full Text](#)
- Frid-Adar M, Klang E, Amitai M, et al.: **Synthetic data augmentation using GAN for improved liver lesion classification**. 2018 *IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. 2018; 289–293. [Publisher Full Text](#)
- Howe B, Stoyanovich J, Ping H, et al.: **Synthetic data for social good**. *ArXiv Preprint ArXiv:1710.08874*. 2017.
- Jeske DR, Samadi B, Lin PJ, et al.: **Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems**. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. 2005; 756–762.
- Kannan S: **Synthetic time series data generation for edge analytics (Version 2)**. *Zenodo*. 2021a. [Publisher Full Text](#)
- Kannan S: **Subar1/synthetic: Synthetic Data Generation**. *Zenodo*. 2021b. [Publisher Full Text](#)
- Kavitha D, Ravikumar S: **Designing an IoT based autonomous vehicle meant for detecting speed bumps and lanes on roads**. *Journal of Ambient Intelligence and Humanized Computing*. 2020; **12**(7): 7417–7426. [Publisher Full Text](#)
- Kumar R, Kumar P, Kumar Y: **Time Series Data Prediction using IoT and Machine Learning Technique**. *Procedia Computer Science*. 2020; **167**:

373–381.

[Publisher Full Text](#)

Li W, Chai Y, Khan F, et al.: **A Comprehensive Survey on Machine Learning-Based Big Data Analytics for IoT-Enabled Smart Healthcare System.** *Mobile Networks and Applications*. 2021; **26**(1): 234–252.

[Publisher Full Text](#)

Liquid Technologies Limited: *Free Online JSON to JSON Schema Converter*. 2001.

[Reference Source](#)

Drapeau M, Bluemle F, Idowu A: 2014.

[Reference Source](#) | [Reference Source](#)

Murshed MGS, Murphy C, Hou D, et al.: **Machine Learning at the Network Edge: A Survey.** *ACM Computing Surveys*. 2022; **54**(8): 1–37.

[Publisher Full Text](#)

Nikolenko S: *Synthetic data for deep learning*. 2019.

[Reference Source](#)

Pedregosa F, Varoquaux G, Gramfort A, et al.: **Scikit-learn: Machine learning in Python.** *The Journal of Machine Learning Research*. 2011; **12**: 2825–2830.

Salim OM, Dorrah HT, El-Kahawy MA: **A novel algorithm to generate synthetic data for continuous-state stationary stochastic process (wind data application).** *2018 Twentieth International Middle East Power Systems Conference (MEPCON)*. 2018; 333–338.

Sun C, Zeng X, Sun C, et al.: **Research and application of data exchange based on JSON.** *Proceedings of 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2020*. 2020; 349–355.

[Publisher Full Text](#)

Tang Z, Naphade M, Birchfield S, et al.: *PAMTRI: Pose-Aware Multi-Task Learning for Vehicle Re-Identification Using Highly Randomized Synthetic Data*. 2019; (pp. 211–220).

Torres DG: *Generation of Synthetic Data with Generative Adversarial Networks*. 2018.

[Reference Source](#)

van der Meulen R: *What Edge Computing Means for Infrastructure and Operations Leaders*. Gartner; 2018.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research