Data Simulation as a Service: The I3T Paradigm on the IoT

D. Karadimas(*), C. Panagiotou, J. Gialelis, C. Chousiadas, N. Panagis, S. Koubias
Industrial Systems Institute, ATHENA Research and Innovation Centre, Patras, Greece
Applied Electronics Laboratory, Electrical and Computer Engineering dpt., University of Patras, Greece

Abstract—**This paper describes the architecture of a new cloud-based tool that focuses on the simulation of data generation from CPS/IoT devices and services. The paper defines the problem statement that the tool addresses and positions itself with respect to the current state of the art. The architectural overview of the tool and the adopted technologies are presented in detail. Finally, the status of the work in progress is presented along with the forthcoming steps.**

Keywords-IoT, CPS, cloud, simulated data

I. INTRODUCTION

The evolution of embedded networked devices has driven the rise of the disruptive technologies of cyberphysical systems (CPS) and Internet of Things (IoT). In parallel, artificial intelligence and machine learning, took advantage of the massive data generation produced by devices sensing and acting on people and their environment, and presented remarkable progress in various application domains (e.g. healthcare, ambient sense & control, industry, etc.) [1].

The intelligence that is introduced in computational systems and services is strongly related and depended on the availability of data. Prior the deployment of an Artificial Intelligence (AI) algorithm in production environment, data scientists require a large volume of data in order to train, test and validate their models [2]. While, the number of devices and services that produce a variety of modalities is increasing day by day, access on these data is not always feasible due to various factors. In many cases, access on these data is not permitted due to privacy regulations. In other cases, the integration with live systems and the usage of their Application Programming Interface (API) to gather the required data demand significant efforts that cost both money and time. Moreover, while some tools that produce data based on existing models do exist, they fail to scale. Consequently, simulating the large volume of data that the numerous CPS and services generate nowadays, in the context of IoT, is not yet efficient. The impact of this lack of modern data simulation tools, reflect on the accuracy and the feasibility of applications that are based on AI and machine learning techniques.

The scope of this paper is to present the architectural design of a web-based, publicly available data simulation tool that addresses the need for "data generation as a service".

In that context, this paper identifies the need for designing and developing tools that address the need for the ever-increasing trend for data. In Chapter 2, problem definition and the contribution of this paper with respect to the literature is manifested. The overall architecture and operational model of the tool is described in detail in Chapter 3. The tool is evaluated through a power consumption web application that is presented in Chapter 4. Finally, the paper concludes with the current status of the work Chapter 5.

II. MOTIVATION

A. Problem Definition

As briefly described in the introductory section of this paper, nowadays, the ever-creasing invasion of AI and machine learning in everyday consumer electronics and web services demand for large volume data consumption. Modern intelligent computational systems owe their evolution to the increase of the generated data coming from cyber-physical systems (CPS), IoT devices and services. However, researchers working in various application domains do not have always this high availability of data. On the other hand, artificial intelligence algorithms cannot succeed their cause unless they are trained and/or applied to large volumes of data. This paper has been motivated by the need for services that provide the necessary tools to generate data of variable volume based on predefined data generation models that the tool is provided by the user.

B. Background

The typical approach for data scientist is to search for public available data repositories that publish appropriate datasets. Such popular repositories are Physionet for medical data [3] and Kaggle, provided by Google, offering data for a wide range of domains. The common path is to handle large csv files with the obtained data and write handlers that feed machine learning libraries written in popular languages such as Python, R, Mathlab, etc. Other approaches try to develop or utilize existing a series of models that use the physical and dynamical knowledge on the domain as well as computational intelligence [4].

III. THE I3T APPROACH

In the context of I3T project (Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments) [5], a novel architecture has been designed for the development of a

978-1-7281-1740-9/19/$31.00 ©2019 IEEE

data simulation tool in order to facilitate researchers, scientists and engineers to feed their web services, and AI systems with generated data, that simulate the behavior of modern CPSs. The high-level architectural overview of the

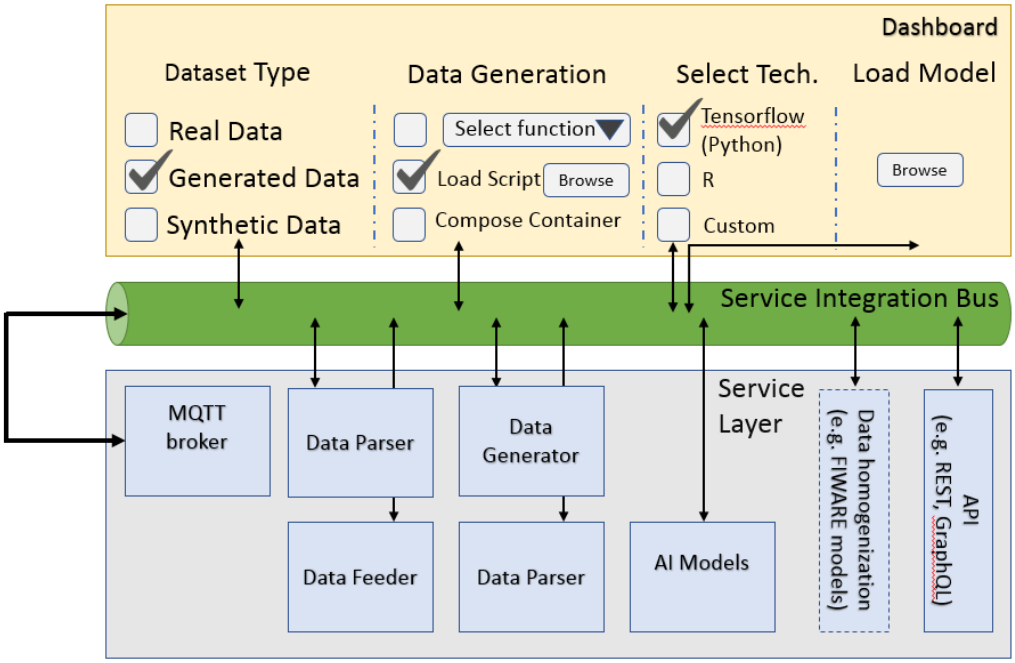

Figure 1: Tool Architecture

tool is depicted on *Figure 1*. The requirements that were defined in order to address the scope of the proposed tool comprise mainly of scalability, technology agnostic, application agnostic, open connectivity.

Scalability: The tool should be able to generate large volumes of data to address applications with similar characteristics (e.g. smart-cities). Therefore, the designed architecture should scale seamlessly to address the required throughput of data generation.

Technology agnostic: As mentioned earlier, data processing methods and AI techniques are implemented in several technologies. Apparently, data experts need to reuse their expertise without spending time on getting familiar with new programming environments.

Application agnostic: The focus of this work was to present an architecture as generic as it gets that tackles a large variety of application domains and not an application specific solution.

Open standards: Finally, in order to facilitate the integration with 3rd party services and infrastructure, the tool should support popular integration patterns based on open technologies.

In that context, the I3T-DaaS (Device as a Service) tool was designed on the principles of microservices architecture [7]. Microservices have gained significant interest lately due to its inherent characteristics such as: high maintainability/testing, loosely coupled services, improved fault isolation, highly distributed environment.

To serve the principles of microservices architecture the docker framework was adopted as the container platform that hosts all the required services [8]. All the individual services of the tool have been isolated in independent docker containers each one dedicated to deliver the functionality of the service it implements. The core of these services comprises of:

1) Dataset generation and feed: As already described, the main contribution of this tool is the ability to generate data with various configurations. The first basic functionality of this service is to give the ability to the user to load files with data and parse them in order to feed the other services such as AI prediction models, APIs, or other 3rd party services. When real data are not available, the user is able to generate simulated data with a variety of data distributions (e.g., normal, Poisson, polynomial, uniform, etc.) and configurations (volume, dimensions, frequency, etc.). For more sophisticated data generation, the user can pass to the service its own scripts in R or Python and generate data based on custom models. Another option that the tool provides is the creation of synthetic data, where simulated data are generated based on existing datasets with real data. Apparently, the user can enhance the performance of the service, by providing its own models and adapt to the characteristics of its application. All this functionality is also depicted in *Figure* 2 and is provided to the user through the web interface of the Dashboard.

2) Data persistence: An additional but optional service, enables the local storage of the data (whether coming from real dataset or models). This service is fed by the generator

service and stores the generated data in databases optimized for timeseries (e.g. InfluxDB [9]), which are ideal for the sensor data of IoT applications.

3) Data homogenization comes in the picture in order to provide a common representation format for the data that the tool generates. Through the adoption of an open standard, the tool can integrate with IoT infrastructures with less implementation efforts and facilitate the optimization of existing services. The homogenization service implements the FIWARE data models to represent the generated data along with the virtual IoT/CSP devices that generate the data [10].

4) API is another key service that enables the extraction of the data (whether real of simulated) using popular http technologies such as the REST web services paradigm. Given the volume of data that could be stored by the tool and made available through the API, query languages such as GraphQL are supported on top of the REST API in order to enable clients to ask and get exactly what they need [11].

5) Another core service that links all the services into a single communication bus is the MQTT [12] broker (as implemented by RabbitMQ [13]). The communication between the I3T-DaaS dashboard and the rest of its services is achieved through message passing as defined by the popular MQTT protocol.

6) Finally, the Dashboard service, is responsible for the configuration of each service and the orchestration of the data flow through an intuitive web interface. The user through the dashboard can define the data generation strategy, to decide whether the data will be stored locally, to enable/disable AI services and expose the data through the API.

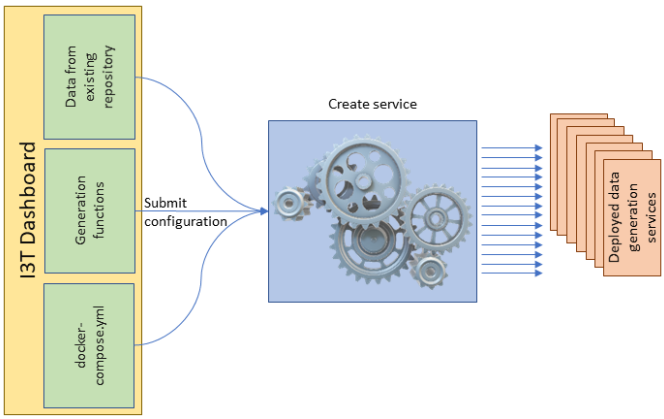

Figure 2: Data generation service

The provided functionality is given as a constellation of docker services where the user can deploy to its own infrastructure and enhance with his custom models and datasets. Based on the volume of load and the required resources the user can scale its deployment though the tools that the docker technology provides. As an example, if the API service of the I3T-DaaS is under high load to serve requests coming from external services, the user can increase the number of the API instances with a single command (e.g. *docker service scale i3t_api=5*). These instances can be distributed to several docker hosts that form a docker swarm without any additional effort from the user to orchestrate and configure these services.

IV. EVALUATION

For the evaluation of the tool, we developed an end-to-end application capable of monitoring the power consumption of a building through a self-intuitive web interface and providing short-term predictions based on the current usage of the domestic appliances. The I3T-DaaS tool had a 2-fold contribution. Initially it was used to support the training and testing process of the AI model. Then, it was used to simulate electric and electronic devices of the building by exposing their consumption.

Particularly, the datasets that were used regard a household's measurements of electric power consumption, taken with one-minute sampling rate over a period of almost 4 years, containing 2.075.259 measurements gathered in 47 months. The data attribute comprises of:

- Time
- Global active power
- Global reactive power
- Voltage
- Global intensity

The data generator service was assigned with the task of data preprocessing and resampled the data from a sampling rate of 1-sample/minute to 30-samples/minute. For that purpose, the python pandas library was utilized [14]. The purpose of this resampling was to reduce the dataset size and predict forthcoming consumption with a smaller sequence of previous values. The data were normalized through the minmaxscaler in order to facilitate the training of our model. Finally, using the Tensorflow [15] tool, the Long Short Term Memory (LSTM) artificial recurrent neural network architecture [16] was trained on these data in order to predict feature power consumption of the devices.

Without any further optimizations, the model presents the performance as described in Table 1, while plots of the predicted values against the ground truth are given in Figure 3.

TABLE 1: MODEL PERFORMANCE

Loss	0.040
Mean Absolute Error (MAE)	0.14
Root Mean Square Error (RMSE)	0.78

The generated data and its predictions have been consumed by the an "Online Platform for the Control of Domestic Appliances". The platform is mainly targeted to electrical

installation service companies. It aims to enable them to provide their customers with a user friendly and efficient experience for controlling their domestic appliances and monitoring the ambient conditions and the residential security. It also gives them the ability to have a user-friendly graphical view of all the networked devices within a space. The status and the controls of each of the devices are placed in a top down view that fully reflects the actual design of the residence. In that sense, the user needs minimal effort to get familiar with the interface since all that he needs is to virtually navigate to his own residence and interact with the avatars of the electric/electronic devices that are installed and integrated to the platform.

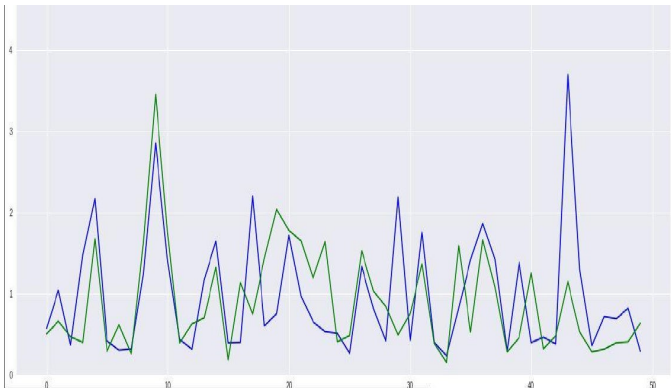

Figure 3: Prediction

Figure 4: Power Consumption graph (history and prediction)

The monitored data (power consumption of domestic appliances) is a mixture of real data coming from devices installed in the lab and simulated data generated by the I3T-DaaS tool and exposed through its REST API. The graphical interface presents the real time values of the consumption of each one, along with the available controls (on/off). Finally, the user is offered a provision of future consumption based on the current patterns given by the trained LSTM model. A view of that interface is given in Figure 4.

CONCLUSIONS

The scope of this use case was not to present an optimized prediction tool for domestic power consumption but to highlight the role of the I3T – DaaS in such workflows. The services of the tool could be deployed in local environment where the user could take advantage of its hardware resources (both CPU and GPU). The tool source code will be available when completed as an open source project through a github repository while, a live web version will be also available for researchers to submit and run their project.

ACKNOWLEDGMENT

This work was undertaken under the “Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments (I3T)” project funded by National Strategic Reference Framework (2014 – 2020) for the Industrial Systems Institute, ATHENA Research Center.

REFERENCES

[1] A. Ghosh, D. Chakraborty and A. Law, "Artificial intelligence in Internet of things," in CAAI Transactions on Intelligence Technology, vol. 3, no. 4, pp. 208-218, 12 2018.

[2] Witten, Ian H., et al. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[3] Goldberger, Ary L., et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." Circulation 101.23 (2000): e215-e220.

[4] Haupt, Sue Ellen, and Branko Kosović. "Variable generation power forecasting as a big data problem." IEEE Transactions on Sustainable Energy 8.2 (2017): 725-732.

[5] https://i3t.isi.gr/en/

[6] https://www.kaggle.com/

[7] J. Thönes, "Microservices," in IEEE Software, vol. 32, no. 1, pp. 116-116, Jan.-Feb. 2015.

[8] https://www.docker.com/

[9] https://www.influxdata.com/

[10] FIWARE http://www. fi-ware. Org

[11] GRAPHQL https://graphql.org/

[12] MQTT http://mqtt.org/

[13] RabbitMQ https://www.rabbitmq.com/

[14] Pandas https://pandas.pydata.org/

[15] Tensorflow https://www.tensorflow.org/

[16] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.