

Study and Development of a Digital Twin for the Simulation of User-Created Automations in a Green Smart Home

Supervisors: Prof. Barbara Rita Barricelli, Prof. Daniela Fogli

Author: Luca Cotti

Abstract

This thesis explores the design and implementation of a Digital Twin (DT) for a green smart home, integrating End-User Development with Machine Learning algorithms. With End-User Development, home inhabitants can manage the behavior of their Internet of Things environments through trigger-action *routines* (Barricelli et al., 2019). The DT allows users to forecast the energy consumption of smart appliances throughout the day and simulate “what-if” scenarios related to the creation of routines operating on the appliances. These simulations assess the effects of the activation of appliances involved in the routines and suggest modifications to save energy. This research is significant as it contributes to the development of a new paradigm for smart and sustainable living based on automatic, user-defined behaviors such as routines.

A hypothetical home was designed as a starting point for the creation of the DT, containing appliances extracted from the GREEND¹ (Monacchi et al., 2014) and UK-DALE² (Kelly & Knottenbelt, 2015) power consumption datasets. These datasets were selected because they provide a large number of appliances and minimize geographic variations in appliance usage patterns. However, they lack information about the *operation modes* of the appliances (e.g., wash cycles for a washing machine or on/off for a lamp), which is required to keep track of their state.

The operation modes, along with their total energy consumption and duration, are identified with an approach inspired by Castangia et al., 2023. Figure 1 illustrates the main steps of the approach. First, a segmentation procedure is applied to the raw appliance data to identify the active states that contain the actual power signatures of the device, called *activations*. These signatures are then standardized and fed to a deep autoencoder model, which learns to reconstruct the operation cycles and encode them into a latent representation. Next, a K-means clustering algorithm is applied to the latent representation to group the operation cycles into different programs of the device. The optimal number value of K for each appliance type is obtained through a grid search that maximizes the silhouette score. Finally, the clusters are mapped to the operation modes of the appliance.

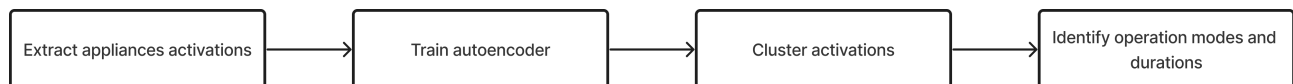


Figure 1: High level procedure for identifying the operation modes of appliances.

¹<https://www.andreatonello.com/greend-energy-metering-data-set/>

²<https://jack-kelly.com/data/>

The DT is implemented as a REST API, separating the DT logic from the user interface. To function, the DT requires the list of appliances extracted in the previous steps, along with a set of routines that control the operation mode of appliances throughout the day. The simulation endpoints of the API specifically evaluate conflicts among routines, focusing on two main issues. Firstly, the routine under creation may cause the power consumption to exceed a maximum limit (e.g., the load capacity of the house). Secondly, an appliance may have conflicting operation modes in the same time interval. To solve these conflicts and reduce energy costs, the DT provides suggestions to either change the routine's start time or disable one or more routines.

To demonstrate the features offered by the DT, a responsive web-based application was developed. It is not meant to be a fully functional interface but a first prototype to show the potential of the system. The application displays a number of statistics (Figure 2), and allows users to simulate the addition of a number of routines (Figure 3), designed to test the behavior of the DT.

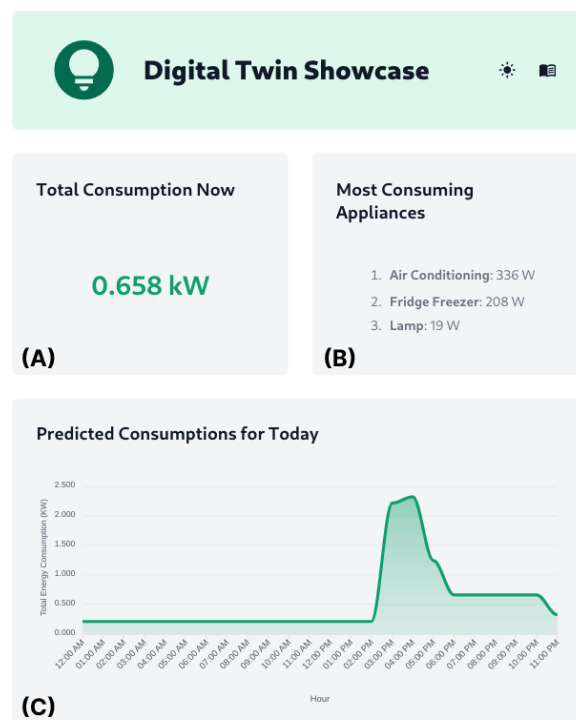


Figure 2: Top part of the frontend application, which shows the (A) total instant power consumption; (B) the appliances that are using most power; (C) predicted daily power consumption.

Figure 3 shows an example of a pre-defined routine in the 'Simulate Routines' section. The routine is named 'Warm up home in the morning' and is set to activate at 07:00 AM. The actions listed are:

- Set the air conditioning to *heat* for 200 minutes
- Set the boiler located in the bathroom to *auto* for 200 minutes

A green button at the bottom is labeled 'Simulate routine addition'.

Figure 3: Example of a pre-defined routine.

References

- Barricelli, B. R., Cassano, F., Fogli, D., & Piccinno, A. (2019). End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software*, 149, 101–137. <https://doi.org/10.1016/j.jss.2018.11.041>
- Castangia, M., Barletta, N., Camarda, C., Quer, S., Macii, E., & Patti, E. (2023). Clustering Appliance Operation Modes With Unsupervised Deep Learning Techniques. *IEEE Transactions on Industrial Informatics*, 19(7), 8196–8204. <https://doi.org/10.1109/TII.2022.3217495>
- Kelly, J., & Knottenbelt, W. (2015). The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2(1), 150007. <https://doi.org/10.1038/sdata.2015.7>
- Monacchi, A., Egarter, D., Elmenreich, W., D'Alessandro, S., & Tonello, A. M. (2014). GREEND: An energy consumption dataset of households in Italy and Austria. *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 511–516. <https://doi.org/10.1109/SmartGridComm.2014.7007698>