Hourly Energy Consumption

Phase5: development part 3

Abstract

Hourly energy consumption profiles are of primary interest for measures to apply to the dynamics

of the energy system. Indeed, during the planning phase, the required data availability and their

quality is essential for a successful scenarios' projection. As a matter of fact, the resolution of

available data is not the requested one, especially in the field of their hourly distribution when the

objective function is the production-demand matching for effective renewables integration. To fill

this gap, there are several data analysis techniques but most of them require strong statistical skills

and proper size of the original database. Referring to the built environment data, the monthly

energy bills are the most common and easy to find source of data. This is why the authors in this

paper propose, test and validate an expeditious mathematical method to extract the building

energy demand on an hourly basis. A benchmark hourly profile is considered for a specific type of

building, in this case an office one. The benchmark profile is used to normalize the consumption

extracted from the 3 tariffs the bill is divided into, accounting for weekdays, Saturdays and Sundays.

The calibration is carried out together with a sensitivity analysis of on-site solar electricity production.

The method gives a predicted result with an average 25% MAPE and a 32% cvRMSE during one year

of hourly profile reconstruction when compared with the measured data given by the

Distributor

System Operator (DSO).

Introduction

Building sector is highly relevant for energy transition since it is responsible for 39% of CO_2 gas emissions and 36% of global final energy use (IEA 2018). It is noteworthy that the adoption of innovative solutions can be strongly affected by the behaviour of building users as tested by Fink (2011), and the time distribution of energy consumption is a crucial aspect to analyze in the view of smart solutions and flexibility enablers as verified by Mancini and Nastasi (2019). Indeed, understanding building energy consumption is the first step for its energy retrofitting (Tian 2013) which will lead to a subsequent performance improvement with a relative reduction of costs and emissions. The claim for just increasing the renewable energy supply cannot alone support the transition toward decarbonization scenarios especially for their non-programmability nature (Loorbach 2010) coupled with the performance gap between design and operation of buildings (Manfren and Nastasi 2020). Smart energy approach is, then, required to link different production and consumption nodes (Rosenbloom and Meadowcroft 2014) keeping in mind that automation and communication in models and reality is fundamental for its success (Tronchin et al. 2018)

Material and methods

From January first 2007, the metering consumption interval is regulated according to Delibera n.181/06 (GU 2006) which was anticipated by Delibera n.19/06 (ARERA 2006), dividing the electric billing measurement in three period called (i) F1 representing the peak period, where the highest consumptions are recorded during the day, (ii) F3 representing the off-peak period, mostly the base load consumption, and (iii) F2 representing the mid-level period, which is the transition interval between peak and off-peak consumption intensities. The subdivisions (Fs) are motivated by a careful trend observation in electricity exchange prices, deeply analysed on the Italian Electricity Market Operator website (GME 2020), which have a very distinguished trend between day and night, and between weekdays (WDs), holidays and weekends. F1 is identified in the time interval going from 8:00 to 19:00 during WDs. F2 is identified from 7:00 to 8:00 and from 19:00 to 23:00 during the WDs and from 7:00 to 23:00 on Saturdays (STs). Ultimately, F3 is recorded from 23:00 to 7:00 during WDs and STs and all day long during Sundays and holidays (both simplified with the acronym SN), that information is schematically summarised in Table 2. The colours used in the table, will be applied always in the same way to guide the reader in easier understanding, accordingly the F1 is exemplified by red, the F2 by yellow and the F3 by green.

As it can be noticed in Table 2, only F1 is recorded always in the same time interval while F2 and F3 change according to the day. Furthermore, F3 is recorded all day long on every holiday day

Machine Learning Modelling

In general, machine learning models work with the dataflow of taking in input features, extracting a relationship with the input feature and the output label, and predicting the future. In our proposed model, the input features of occupancy rate, seasonality, and datetime are given to the deep neural network model as input features. The deep learning models output is essentially the estimated value of energy consumption as illustrated in Figure 5. As indicated in the previous section, the synthetic load generator was used to generate the dataset in this pattern, and then the full set of data was then fed into the proposed deep learning model. The learning requires two stages, the training stage to create the prediction model and the testing stage to verify the prediction model's prediction performance.

Furthermore, the Python programming language with Tensor flow and Keras libraries was used to develop the MLR, XGB, and shallow/basic ANN models, and the proposed deep neural network model. A different number of hyper-parameter tuning approaches were included in the shallow/simple or conventional ANN model to obtain the proposed deep neural network model. Results indicated that after the hyper-parameter tuning, the prediction accuracy of the model had improved significantly.

Proposed Deep Neural Network Model

A machine learning model's performance is heavily dependent on its hyper-parameters

and in general, the hyper-parameters are tunable, and finding an optimised value for these

parameters can directly influence the performance of the model [23]. It is essential to understand that this study focuses on optimising a shallow ANN model's hyper-parameters

to obtain a more accurate and useful deep learning model. On the other hand, hyperparameters are external parameters that are set by the operator of the network [24]. For

example, there are two types of hyper-parameters: Hyper-parameters related to neural network structure (number of hidden layers, dropout, activation function, etc.)

program

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
print(summary_stats)

# SUPPORT VECTOR MODELLLING
print(BLUE + "\nMODELLING" + RESET)
# Reduce the dataset size for faster training
df = df.sample(frac=0.2, random_state=42)
# Split the data into features (Datetime) and target (AEP_MW)
X = df[["Datetime"]]
y = df["AEP_MW"]
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42 )


# Preprocess the features (Datetime) to extract the day of the year
```

```python
plt.figure(figsize=(10, 6))
plt.hist(
    df["AEP_MW"],
    bins=100,
    histtype="barstacked",
    edgecolor="white",
)
plt.xlabel("AEPMW")
```

```python
plt.ylabel("Frequency")
plt.title("Histogram of MEGAWATT USAGE")
plt.show()
```

```
DATA CLEANING
Missing Values :
Datetime    0
AEP_MW      0
dtype: int64
Duplicate Values :
0
```

```
DATA ANALYSIS
Summary Statistics :
                            Datetime          AEP_MW
count                         121273   121273.000000
mean   2011-09-02 03:17:01.553025024    15499.513717
min              2004-10-01 01:00:00     9581.000000
25%              2008-03-17 15:00:00    13630.000000
50%              2011-09-02 04:00:00    15310.000000
75%              2015-02-16 17:00:00    17200.000000
max              2018-08-03 00:00:00    25695.000000
std                              NaN     2591.399065
```

```
MODELLING
Mean Squared Error: 6758395.805638685
R-squared: 0.00270160624748228
```

Conclusion

The presented results provide a starting point for the research community as it tries to accurately estimate the amount of energy consumed in software development leveraging the benefits of ML. However, there is still room for improvement in the proposed approach above. The challenges faced by the research community have been presented and the existing energy consumption measurement tools have also been discussed.