**Microwave Detection**

Introduction

In this description, our topic of discussion will be the disaggregation of microwave ovens. Before we start, we should establish that the report about water heater detection is meant to be read first because since the general idea remains the same in this report we will touch on the differences between this implementation and the one about water heaters, while making small references to similarities when they are worth pointing out.

A graph with green lines

Description automatically generated**Scripts developed and deployed**

* getEvents.py ***[1]***
* uploadToTB.py ***[2]***
* delOverlapping.py ***[3]***

It should be obvious that they are the core architecture has remained unchanged, even to the names of the scripts.

A graph showing a loss of a graph

Description automatically generated with medium confidenceThat means that the entirety of the paragraph describing how the scripts work together to achieve the desired result is identical in this iteration. Also, both ***[2]*** and ***[3]*** have maintained the same functionality, with only some small changes that are not worth mentioning.

In the next section we will study the elements that actually differentiate the two iterations and the reasons why each change has been applied.

**So, what is new?**

As goes with the disaggregation for any different device, the crux of the implementation is the set of rules that we establish in order to make the device in question detectable. It should be obvious that all the rules applied here will be different than the ones written for water heaters, since the qualitative graph depicting the consumption of a microwave is fundamentally different from that of a water heater.

As we have established, a water heater’s consumption is expressed in a cartesian plane, resembling a square pulse, and maintaining a somewhat stable value while the device operates.

In contrast, a microwave oven’s footprint will be expressed on a cartesian plane of consumption by time as a series of peaks following one another at a rapid pace. So, there are very substantial fluctuations whereas in a water heater’s consumption graph, they should be mostly negligible (see fig 1 and fig 2).

Fig. 1: The general form of a microwave oven’s event

Fig. 2: The general form of a water heater’s event

**The set of rules**

* Events are identified and deemed as either ON or OFF.
* An event is considered ON when: **1)** its first value is both smaller than the value with the same index -2 and -4, **2)** its sixth value is smaller than the one before the last of the previous event.
* An event is considered OFF when **1)** its first value is both smaller than the value with the same index +2 and +4 **2)** the value with index of the last of the previous event +5 smaller than the third value of the current event. That method is safe because given the frequency at which measurements are made, an event should contain many more values, so the eight first should always be ascending for a water heater.
* Consecutive events of alternating types (e.g. ON, OFF, ON) that occurred closely to each other, namely 90% of their duration, are grouped together.
* Consecutive groups that occurred one minute or less apart will be merged. That provides a tolerance for any events that were missed, possibly resulting in two consecutive events of the same type. With this rule that issue is dealt with.
* Finally, any events lasting for 30 seconds or less, that made it through the rules above, are discarded since they are much more likely to just be false positives, because of other devices operating at that time, in contrast to an actual operating cycle of a microwave. Worst case scenario? A 30 second event is lost.