**Uma imagem com texto

Descrição gerada automaticamente**

**Computational Intelligence for the IoT**

**Project 1.2**

**Classification using**

**Fuzzy Systems and NN**

**Professor:**

João P. Carvalho

**Students (G2) :**

Aurora Nora | 93573

João Lopes | 93584

In this report we will explain how we developed a model using a Fuzzy System with the purpose of knowing if the number of students inside a lab is above the limit (2 students), based on data received from several sensors inside that specific room.

Initially, we analyzed the given dataset and the problem that we would have to solve in the development of the Fuzzy System. We also started thinking about which features would make sense to use this time, as we can’t have so many inputs in fuzzy systems because of the complexity.

After exploring the possibilities and thinking on which would work out better in the solution of the problem, we agreed on using the light features (“S1Light”, “S2Light”, S3Light”) to try estimating the number of lights on and consequently the number of person in the room, also the CO2 variation trough the day and it’s the acceleration so we can know if the CO2 is stable or not and also we used the date and the hours to divide the daytime in “day” and “night”. We noticed that we would have to use the combination of “S3Light” and “CO2” and “Time” to guess if the weather is “sunny” or “cloudy”.

**Lights:** Sum of the lights.

* Composed by: “S1Light”, “S2Light”, S3Light”
* If it’s “night” [>700] -> lab overcrowded | [<700] -> lab undercrowded
* If it’s “day” and “sunny” [>1000] -> lab overcrowded | [<1000] -> lab undercrowded
* If it’s “day” and “cloudy” [>700] -> lab overcrowded | [<700] -> lab undercrowded

**Daytime:** Separates the day in two parts – “day” and “night” by using the columns of date and the hours.

* Day - 7am-7pm. | Night – 7pm-7am.

**CO2:** The variation of values of the levels of CO2 in the lab classroom (if it’s decreasing fast, decreasing, stable, increasing, increasing fast).

* Decreasing Fast [-1, -1, 0.5]
* Decreasing [-1, 0.5, 0]
* Stable [-0.3, 0, 0.3]
* Increasing [0, 0.5, 1]
* Increasing Fast [0.5, 1, 1]

The normalizations was made z = x / max(|min CO2|, maxCO2).

We ended by not using this feature because ….

**Weather:** It’s composed by the values of Daytime, S3Light, CO2.

* Sunny – SL3Light > 170 | CO2 < 400
* Cloudy – SL3Light < 170 | CO2 < 400

**Fuzzy Rules:**

1. Night AND lights < 700 -> UnderCrowded
2. Night AND lights > 700 -> OverCrowded
3. Day AND sunny AND lights < 1100 -> UnderCrowded
4. Day AND sunny AND lights >1100 -> OverCrowded
5. Day AND cloudy AND lights < 700 -> UnderCrowded
6. Day AND cloudy AND lights >700 -> OverCrowded