Task Description

The dummy data file provided with the work package provides you a time series data in csv format where each entry has **a timestamp, id, x_position, y_position, unique_id** and **sensor_id**. It is a data of a scenario where some objects are moving around and detected by a sensor network. Each object might be detected by more than one sensor.

Task is to write an algorithm that reads the data sequentially (one by one) according to timestamp and cluster the data coming from different sensors (use sensor_id) and save the fused data to a new csv file where each entry should have **f_timestamp, f_id, cluster_data, f_u_id.** The data in the new csv file must be updated sequentially and should not be done all at once.

You have to follow following rules/constraints while doing the above task:

1. **unique_id** is a universal ID, that means if 2 objects have same **unique_id** irrespective of other details, they are the same real objects.
2. It is not necessary that **unique_id** will be visible always. If the **unique_id** is 0 that means it is not known.
3. x_position and y_position for the same object from 2 or more different sensors at the same timestamp might be different
4. **id** is specific to individual sensor only and has no relation to the id from another sensor. The sensor assigns a random ID to each detected object. At same timestamp values, if one sensor is showing 2 or more different objects with different id then these objects are different for sure
5. Clustering of data should be performed based on the positions (x, y) and distance metric but at the same time keeping in mind the rule number 1,2 and 3. The maximum distance threshold to fuse any data together is 2 meters.
6. If an object with some unique_id from one sensor is clustered together with object from another sensor (without some unique_id) at a similar timestamp, then the two objects should have the same unique_id. So the object without the unique_id in this case should be assigned the same unique_id.
7. **cluster_data** is a list of lists, where the the individual list inside the main list contains data in the format [x_position, y_position, sensor_id]. These values are from the first csv file that are clustered together based on above rules.
8. **f_u_id** is the same unique_id (single value) from one or more entries that are clustered together at this instance
9. **f_id** should be randomly assigned to the clustered data irrespective of the fact is f_u_id is available or not and this f_id should remain same for the same clustered object
10. **f_timestamp** is the value of a common timestamp from the entries that were clustered together. You can use averaging of timestamp wherever required

11. Reduce the time complexity of the algorithm to as minimum as possible. Time of execution of code will also be taken into account by us.

The given task must be completed in maximum 5 hours. You have the first 30 minutes to understand the task and ask questions related to the task. After 30 minutes, questions will not be answered and you can assume anything which is missing or which might not seem correct to you.

You can submit the results before 5 hours also if you want. A faster time of submissions will be considered during the evaluation.

You can also submit incomplete task. It is not necessary to submit complete task. A comparative evaluation is done and in past candidates with partial results have also been accepted for next round of evaluation.

The final format of submission should be a private GIT repo which you should share with l.bisht@sentics.de