Algorithm 1 Learner. An observation consists of a list of values, with the format: (time,value). The behavior of an observation is expressed as a list of mathematical equations, which are afterwards incorporated to a graph model *lm*. All inputs are given by the user.
Inputs: Observed data *observation*, buffer size *bufferSize*, time step *timeStep*, empty learned model *lm*.
Output: Learned model *lm*

```
1: procedure LEARNER(observation)
2:     trace ← parseObservation(observation)
3:     buffer ← bufferInit(timeStep, bufferSize, trace)
4:     equationTrace ← fit(timeStep, buffer, trace)
5:     if lm.isEmpty() then
6:         for each equation ∈ equationTrace do
7:             addEquationToLearnedModel(equation)
8:         end for
9:     else
10:        directSuccessors ← getDirectSuccessors(lm.initialLocation)
11:        for each equation ∈ equationTrace do
12:            directSuccessorsDistances ← getDistances(directSuccessors,equation)
13:            lastModifiedNode ← addEquationToLearnedModel(directSuccessorsDistances)
14:            directSuccessors ← getDirectSuccessors(lastModifiedNode)
15:        end for
16:    end if
17: end procedure
```