

## 0.1. Theoretical Formalization

Given a set of agents  $N = \{0, 1, \dots, n\}$ , a fixed set of features  $D = \{f_1, f_2, \dots, f_d\}$  and a number  $T$  of timesteps with  $T \in \mathbb{N}^+$ , our problem consists in predicting a binary label  $y_t \in \{0, 1\} : t \in [0, T]$  given a data sample

$$\mathbf{x}_t = (x_0, x_1, \dots, x_d)$$

with  $d$  being the number of features and  $x_i \in \mathbb{R} : i \in [0, d]$ .

To analyze the prediction from a broader view we can say that our model predicts a set of labels  $\mathbf{y}_i = (y_1, y_2, \dots, y_T)$ , with  $T$  as the total number of timesteps in the time series, given a set

$$\mathbf{X}_i = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$$

of data samples  $\mathbf{x}_t : t \in [0, T]$  with  $d$ -dimensions.

In the end our data is collected in the form

$$\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)$$

where  $\mathbf{X}_i \in \mathbb{R}^{T \times d}$  is a time series of size  $T \times d$ . The goal of our work is to experimentally find which subset  $D' \subseteq D$  provides the best performance for the task of predicting  $y_t$  given  $\mathbf{x}_t$  with  $t \in [0, T]$ .

## 0.2. Problem Setting

The problem of FDD in swarms robotics can be analyzed in different scenarios with different approaches. As we have seen in Section ??, it is possible to observe the task execution from different points of view. Based on which interpretation we choose, we have to analyze some aspects instead of others. As explained in Section ??, the agents we use offer several kinds of sensors from which we can retrieve different forms of attributes or features. In our approach, we choose to analyze the problem from an external point of view with a centralized controller completely detached from the agents. This approach is coherent with the swarm concept since it does not require continuous communication with a central controller (the swarm paradigm allows only broadcast messages directed to other agents in the swarm). In other words, we do not consider information that directly comes from the robot sensors, but we consider only information that can be gathered from an external observer.

The data collection procedure we propose starts from the injection of faults in simulators, proceeds on extracting the data of the simulation execution from the point of view of an external observer, and then elaborates the data to build the multivariate time series. In real-world scenarios of fault detection, or more generally anomaly detection, it is quite rare to have the chance of simulating the environment under analysis, nonetheless, simulations might be costly in order of time or money for equipment. Being able to simulate can be considered an advantage but has to be used with caution. Even if we can simulate an arbitrarily large number of agents and repeat the simulations several times, we have to keep in mind that the simulations must reflect real scenarios and not detach too much from concrete possible situations.

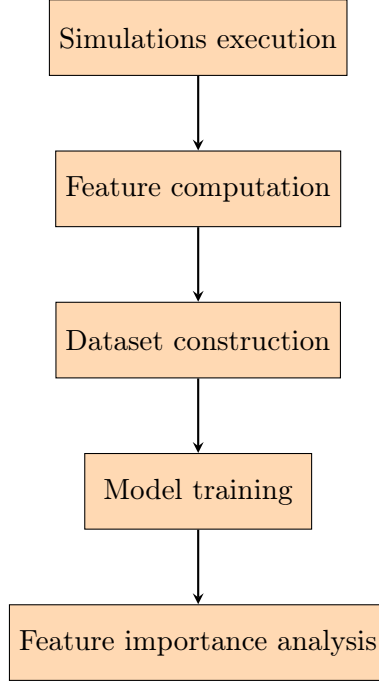


Figure 1: Fault detection algorithm summary.

In Figure 1 we can see a high level summary of the procedure described in this thesis. The project starts from the collection of several simulation log files with the positions of the robots which are then parsed to build the multivariate time series with the features listed in Section ???. Once the features are computed, we proceed on building the datasets needed to retrieve the model results and execute the training phase. When the model completes the training phase, we compute the results of the fault detection and the feature importance values on the test set. The outcome of the fault detection procedure are explained in Section ??.

In this project, we have not analyzed all the possible faults with all the possible combinations of parameters, instead, we have focused on a subset of realistic scenarios easy to implement. Our ultimate goal is to build a dataset that can encapsulate some conditions that reflects real-world situations while being easy to simulate. The choice of faults types and length of simulations is also a consequence of restricting the size of the problem: as usual for problems with several variables, they can quickly explode and become unfeasible to analyze.