

Robots are a concrete companion in our everyday life and constitute a fundamental gear in the nowadays production system. Since the introduction of the first industrial robots in the middle of the 20th century [? , pg.3-5], robots are a worldwide widespread production basic instrument. The rapid growth and diffusion of robot technologies have been driven by the need for automation and acceleration of human manufacturing capabilities. The propagation of robotic systems and the need of integrating multiple robot-operated machineries guided the birth of multirobot systems. This new approach leads to a new interpretation of the robot as an agent of a more complex system. With the development of the multirobot systems interaction capabilities, swarm robotics has become a concrete reality representing a fundamental component in nowadays process industries. It is possible to observe swarm technologies in industrial fields as managing warehouses [? ], in environment preserving tasks like ocean-skimming and oil removal [? ] or for farming activities [? ]. Swarm robotics is defined as “the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among the agents and between the agents and the environment” [? ]. A swarm of robots can be considered a robust system thanks to its inherited properties: redundancy, naturally distributed system, and individual simplicity [? ]. However, it has been demonstrated that, together with its advantages, swarm robotics exhibits high susceptibility to system failures due to individual failures [? ]. To reach the goal of long-term autonomy [? ], it has become a compelling need to build and deploy safe and reliable products. Modern days approaches have studied a collection of techniques to detect and diagnose faults to decrease downtimes and to be able to ensure safety and reliability for the entire system lifetime. Our approach focuses on a centralized machine learning approach for fault detection and, more precisely, we try to understand which are the features to analyze to perform fast and reliable fault detection.

Autonomous agents can be either software or physical agents, both representations can be analyzed as an entity that perceives and interacts with the environment. In the world of swarm robotics, the individuals that make up the swarm robotic system have to be autonomous with a physical embodiment in the world and they have to physically interact with the environment [? ]. The perception of the environment happens through the acquisition of data from several sensors which varies from different robots and different tasks, while the environment interaction happens through the activation of different actuators. Due to the physical limitations of sensors, the swarm agent is unable to perceive the entire environment and all of the other agents; the partial observability of the environment is one of the characteristics of swarm robotics that poses itself both as an advantage, to limit the size of analysis space in the single agent, and as a disadvantage, to limit the agent sensory capabilities.

The use of robots becomes very useful when it is necessary to execute a task that is unfeasible or too demanding to humans. These kinds of tasks are defined with the four D’s: too Dangerous, too Dull, too Dirty, and too Difficult [? ]. In production environments such as warehouses or industrial processing plans, robots reliability has become a key aspect to ensure security and decrease losses in terms of time and resources. Even if the deployed robots guarantee a declared level of reliability, it has become compulsory to perform fault detection and diagnosis to identify failures as quickly as possible. Being able to promptly detect malfunctions enables faster recoveries and lower downtimes. Fault detection and diagnosis in swarm robotics is already a deeply investigated subject with plenty of re-

search possibilities for new findings and improvements. The main challenges we face are posed by the swarm paradigm, this approach offers a lot of analysis surface but presents technical complications like the size of the analysis space and the computational time required to identify an anomaly.

Fault detection is a field of study that resembles the field of anomaly detection, they both share common technicalities and approaches. It is possible to find studies that refer to model-based fault detection that dates back to 1997 [?] and more recent data-driven approaches [?]. Modern days approaches focus on machine learning techniques [?] and aim to perform fault detection with unsupervised methods exploiting deep learning [?]. All the machine learning and data-driven approaches offer a vast field of techniques to test, however, in front of all these possibilities, it comes in handy to analyze which kind of data and which features are useful to analyze to obtain significant performances.

Our project is based on the analysis of task executions from two different simulators, namely ARGoS and RAWSim-O, and the prediction performance of a Gradient Boosting model. The ARGoS simulator is used to simulate the tasks of flocking and foraging while RAWSim-O simulates the execution of a warehouse management situation. We injected the faults in the simulator's core code and then we have collected different task executions. The centralized approach of this work refers to the point of view from which we observe the swarm behaviors. In our work we assume there is an external observer which has to control each agent in the swarm. The activity of the external controller consists in identifying each agent and computing its position with respect to a local positioning system. This assumption allows us to detach from the concept of retrieving data directly from the agent itself, allowing us to detect any kind of fault. We use the data collected from the simulations to compute complex features and build a multivariate time series. The model in consideration is trained on datasets composed of single timesteps from chained multivariate time series with an equal number of nominal and fault samples. The goal of our research is to analyze the informativeness of raw data like position, speed, orientation, and complex data like position entropy, number of neighbors, and area coverage to name a few. Our experiments show that the best features are the ones that are most correlated with the type of malfunctioning implemented and collect more information about the injected fault. In particular, we can observe that adding more features slightly improves the model performances but requires more computational time. In conclusion we confirm that it is important to use background knowledge and narrow the exploration space to the specific task we are analyzing. The set of features we use might improve the performances of the model allowing it to analyze more aspects of the swarm robots behavior, but it is important to check if significant results could have been reached without most of the analyzed features.

The thesis is organized as follows.

In Chapter ??, we describe the current state of the art in the field of fault detection for swarm robotics and present the theoretical background of the instruments we have used to develop our work. Chapter ?? presents the theoretical formalization of our approach and describes the overall environment in which this thesis is constructed. Chapter ?? introduces the technical instruments we have used to simulate a swarm robotic environment, it proceeds on presenting a general overview on how the swarm robotic tasks work,

and, in the end, it explains the modification we have implemented to collect the data from the simulators. In Chapter ?? we present the precise settings we have used to execute the swarm simulations and we explain in detail the procedure to compute the features used to perform the fault detection. In Chapter ?? we show the graphs of the the elaborated data and comment on how they can influence the model performances. We proceed then on describing the results obtained from the model and how they can be interpreted. Chapter ?? concludes the thesis by showing the results obtained from our experiments and what are the final consideration we can derive from them. We finish the chapter by analyzing future development opportunities for our work, listing some aspects that can be further explored, and proposing more advanced technologies that can improve the performance and the results of this work.