

Open Problems in Robotic Anomaly Detection

Carnegie Mellon University

Software Engineering Institute

Ritwik Gupta

Copyright 2018 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

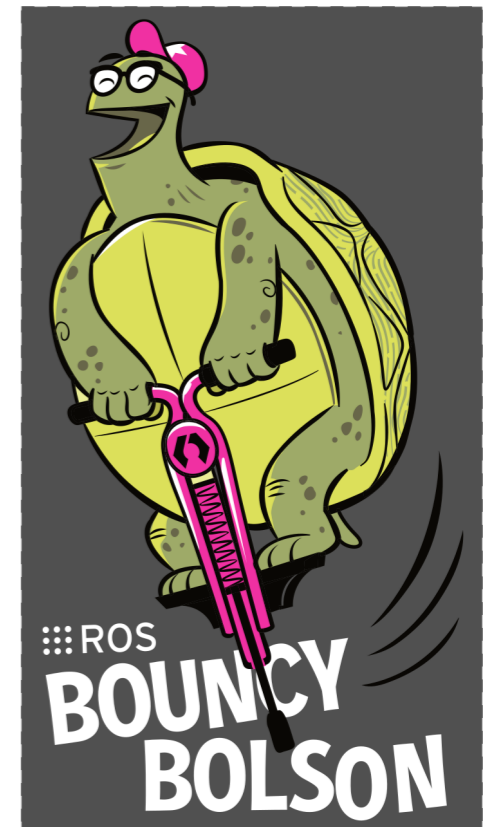
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM18-0856

Abstract



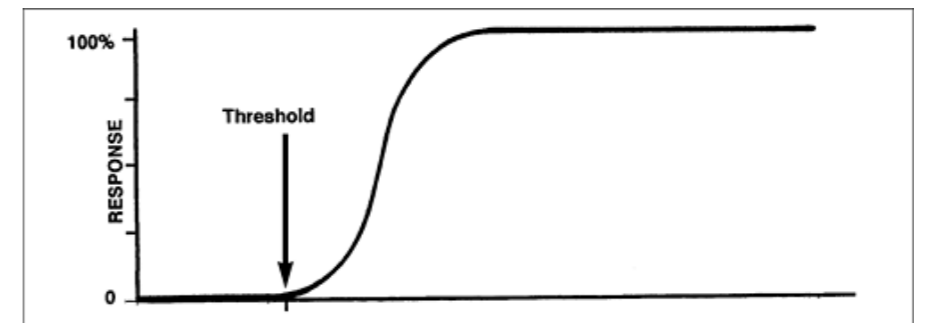
+



Non-malicious faults present many false alarms

Not all failures are anomalies, and certainly, not all anomalies are failures

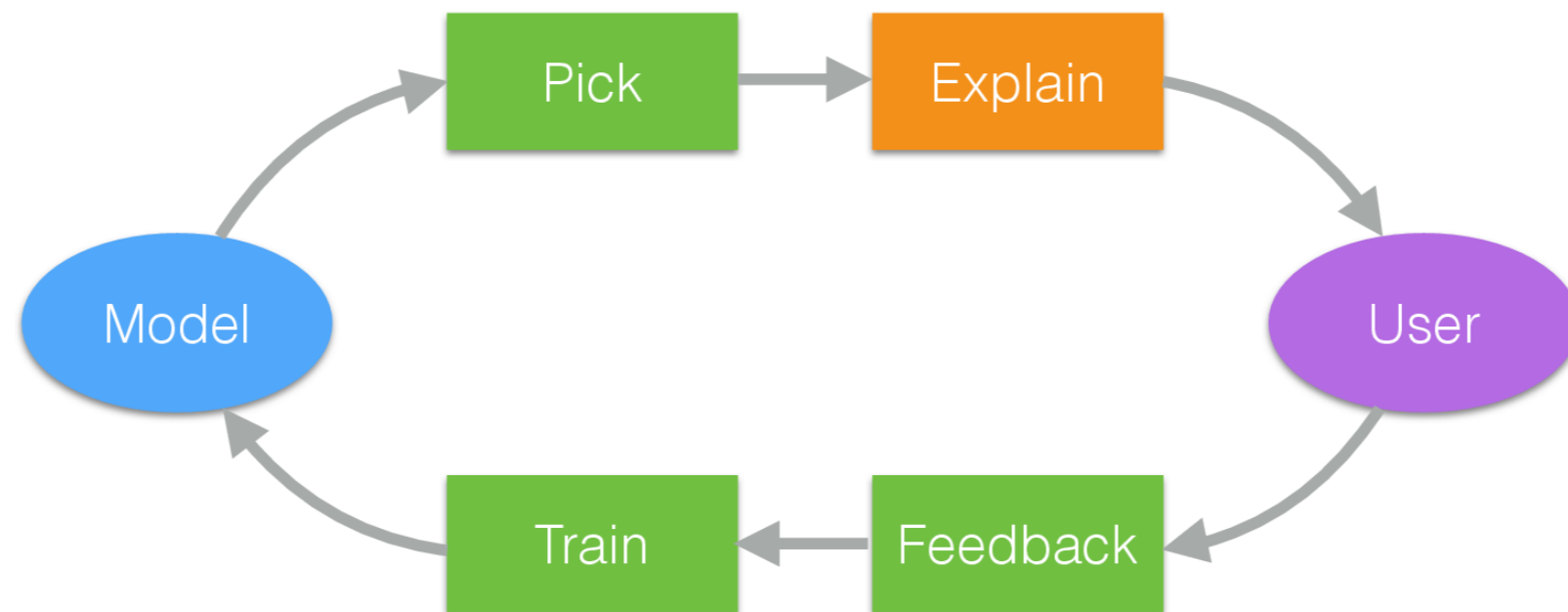
↪ A robot could behave anomalously often without ever failing!



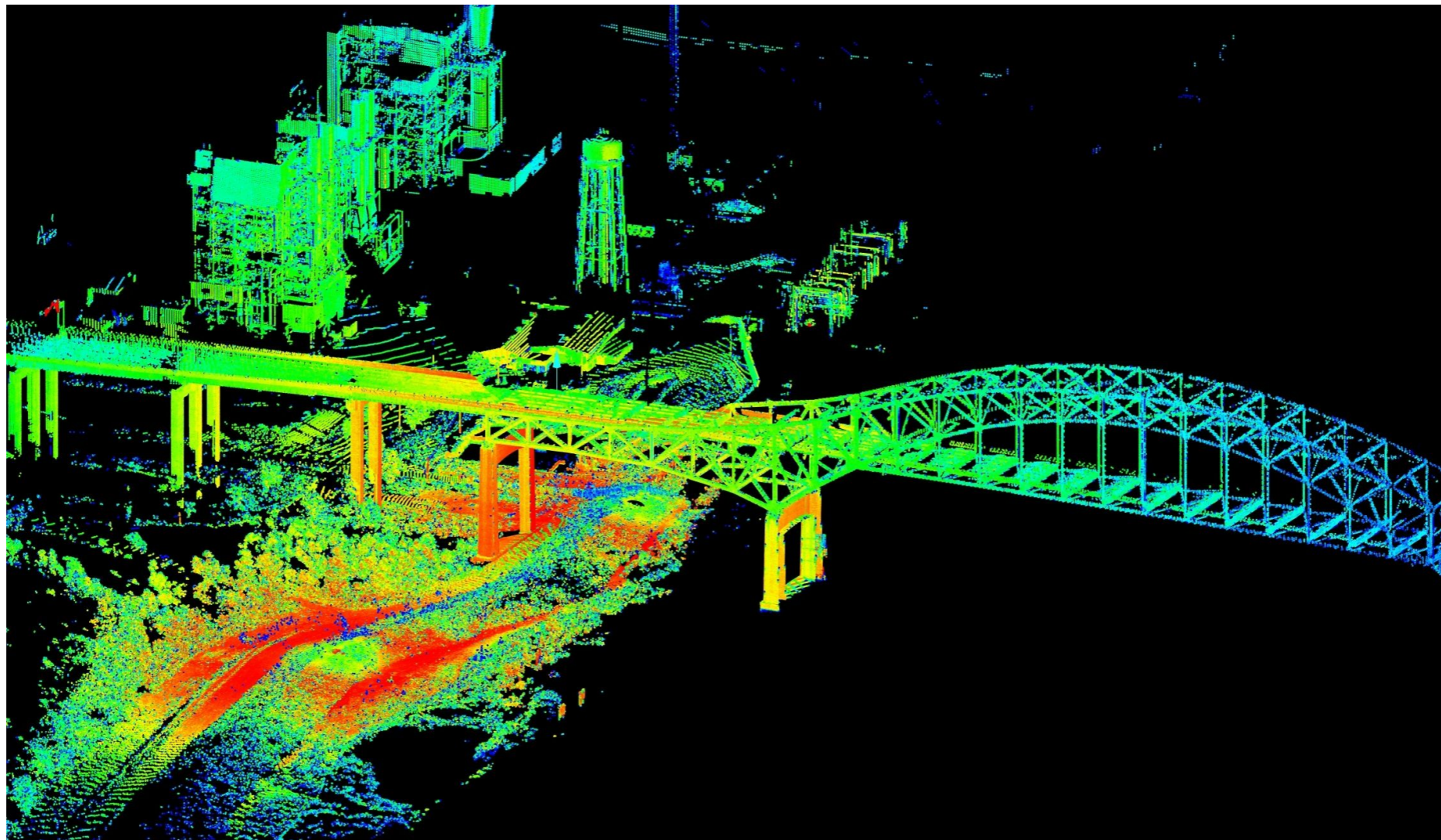
Non-malicious faults present many false alarms

Not all failures are anomalies, and certainly, not all anomalies are failures

↪ A robot could behave anomalously often without ever failing!



When is invalid data anomalous?

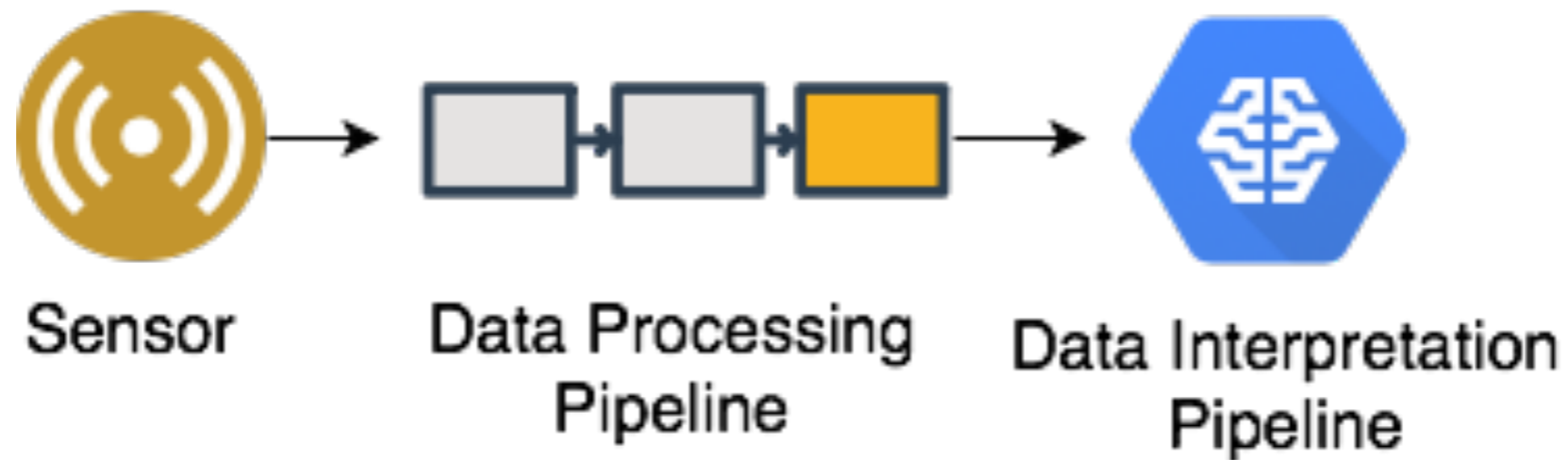


When is invalid data anomalous?

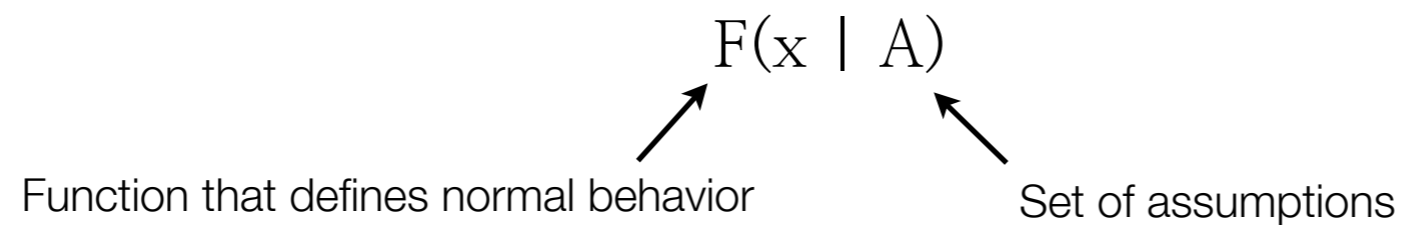
**Data is never anomalous;
interpretations are**

**Data can be flawed
given a static interpretation
framework**

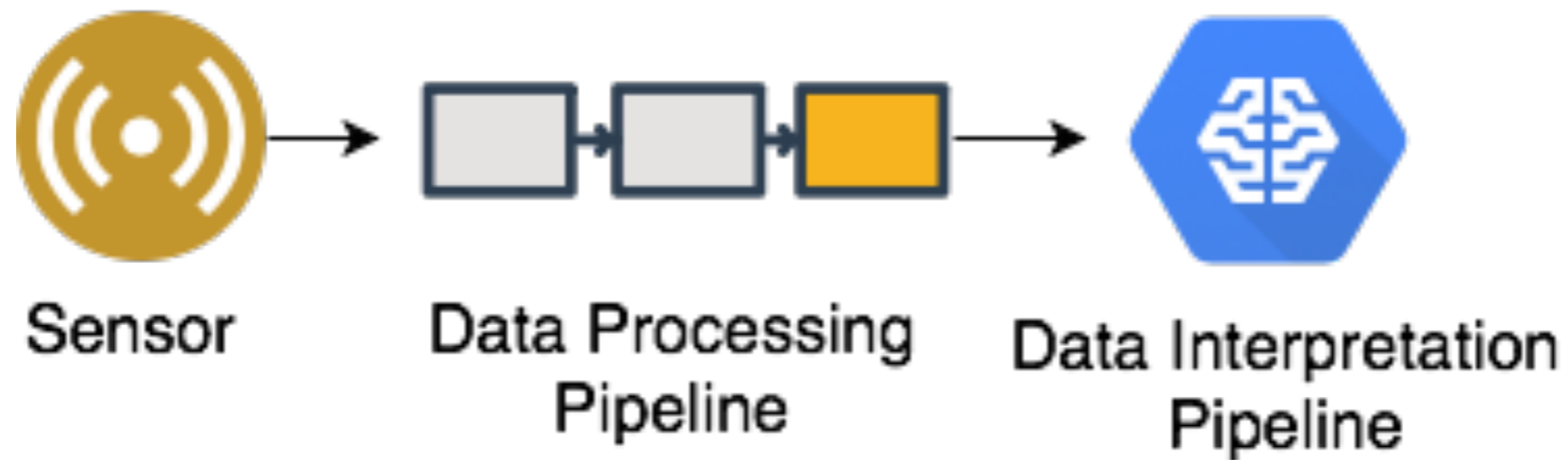
When is invalid data anomalous?



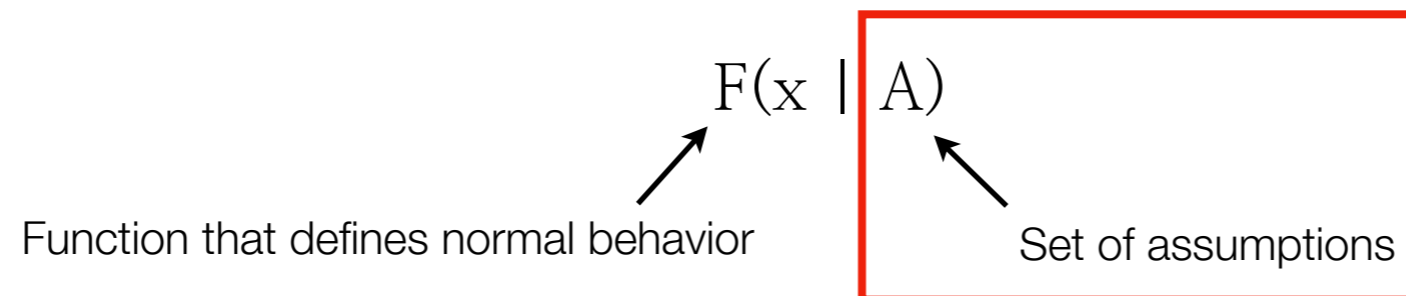
Invalid data \neq Anomalous behavior



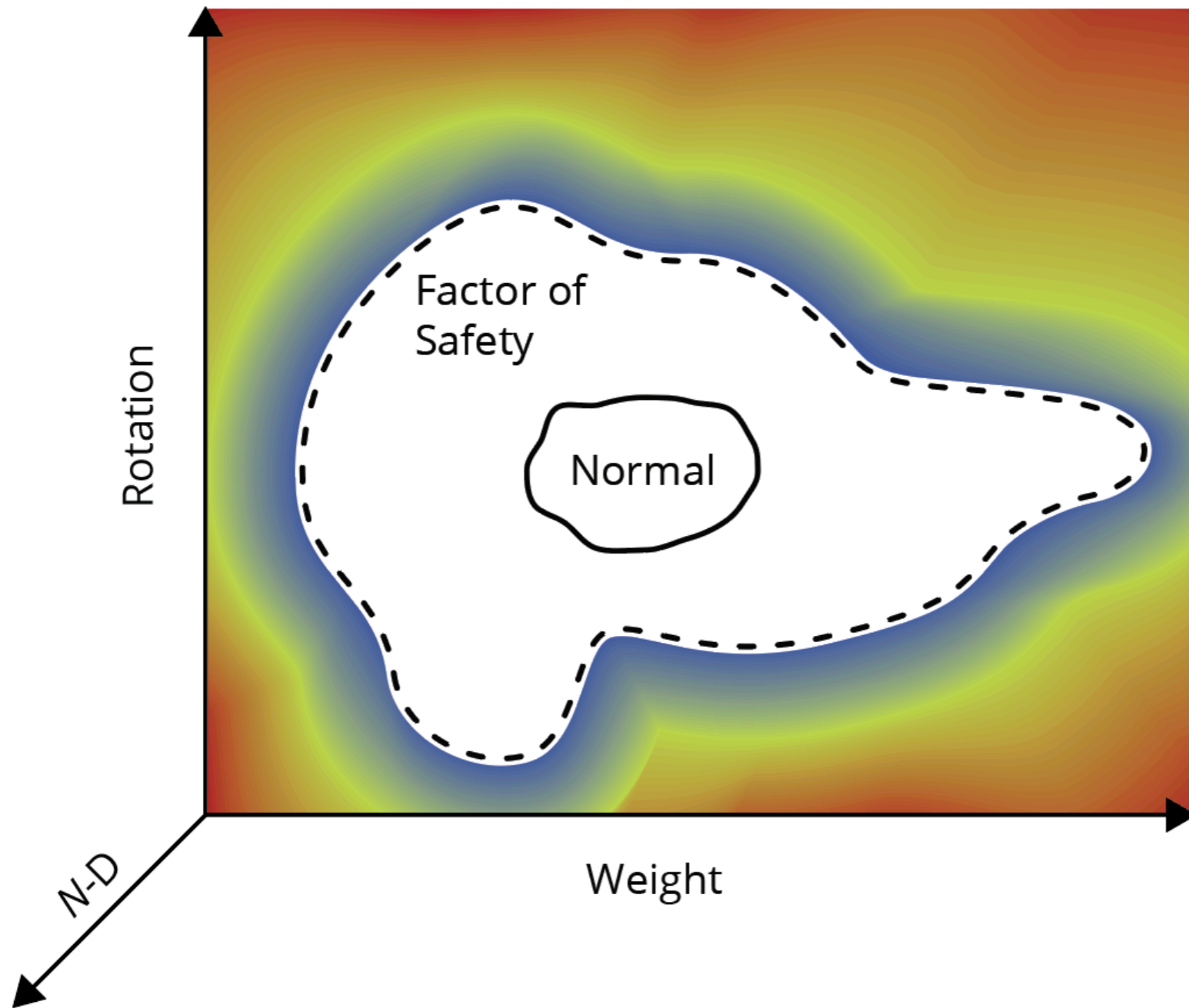
When is invalid data anomalous?



Invalid data \neq Anomalous behavior



Intentional anomalous behavior and emergency stops



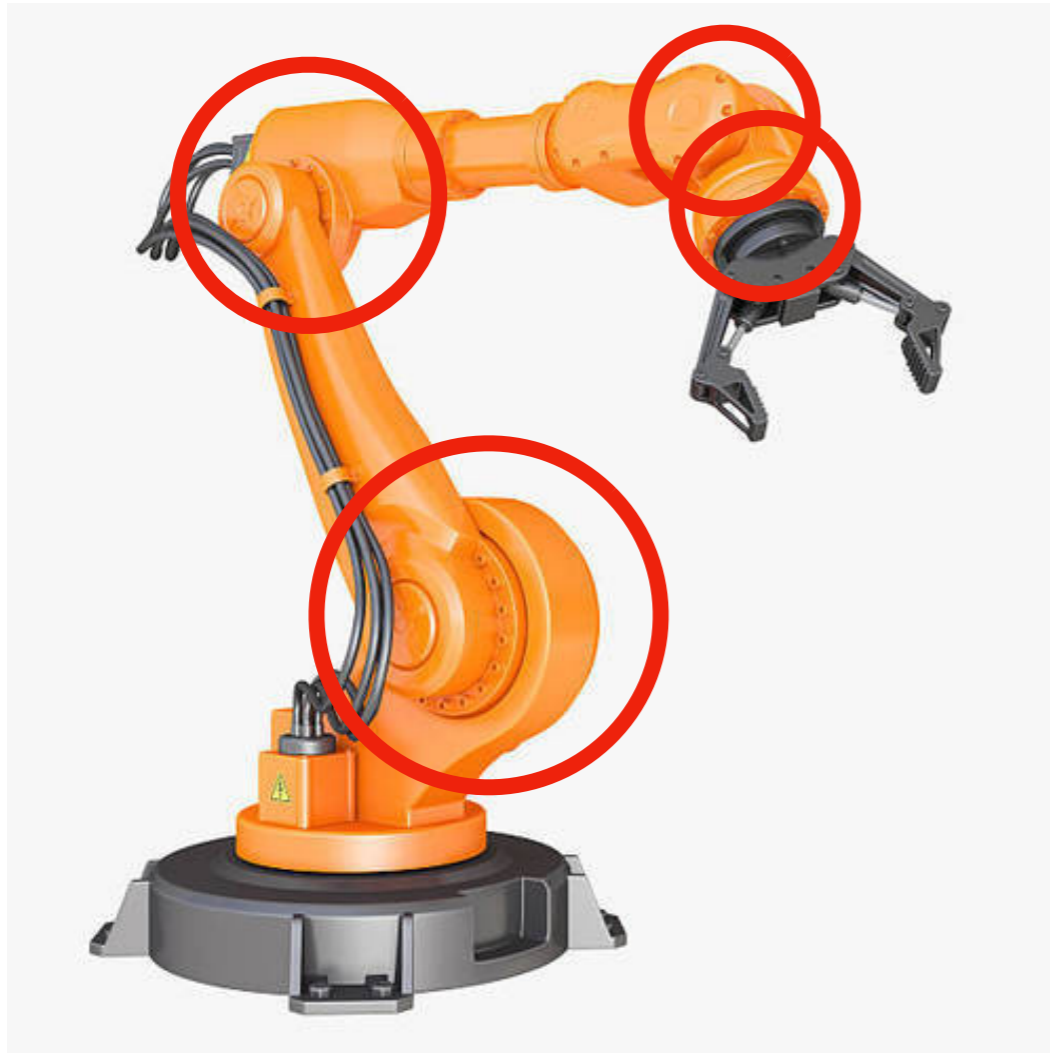
Given some state $\phi \in OC$,
when does it represent
anomalous behavior?

Hierarchies of systems with shared functionality

A robot is defined as:

- a collection of k nodes $V = \{v_1, \dots, v_k\}$, where some nodes are connected by directed edges $E = \{(v_i, v_j)\}$ variously representing physical anchoring, energy flow, or information flow of various kinds,
- the graph is defined as $G = (V, E)$,
- nodes can be grouped in the form of $\{v_x \mid f(v_x)\} \exists v_x \in C$, where $f(x)$ represents a predicate function that returns true if v_x has a certain functionality, and C represents the overall set of all groups in the robotic system,
- and v_x is a member of only one subset of C

Hierarchies of systems with shared functionality



Composability!

Behavior of nodes V :

$$B = [b_1, \dots, b_k], \text{ where } |B| = |V|.$$

Vector of constants $\Phi = [\alpha_1, \dots, \alpha_k], |\Phi| = |B|.$

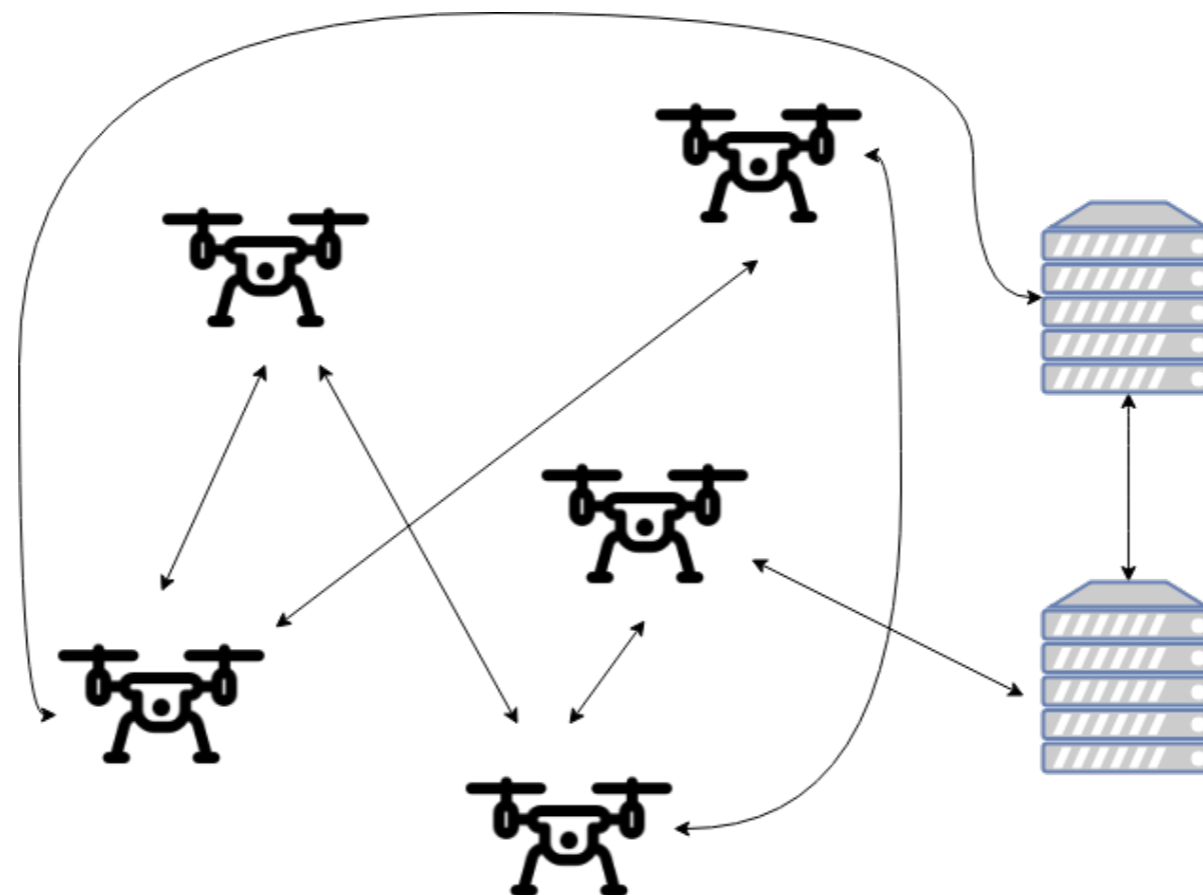
Linear composability is then defined by:

$$\Phi^T \cdot B = \alpha_1 b_1 + \dots + \alpha_k b_k$$

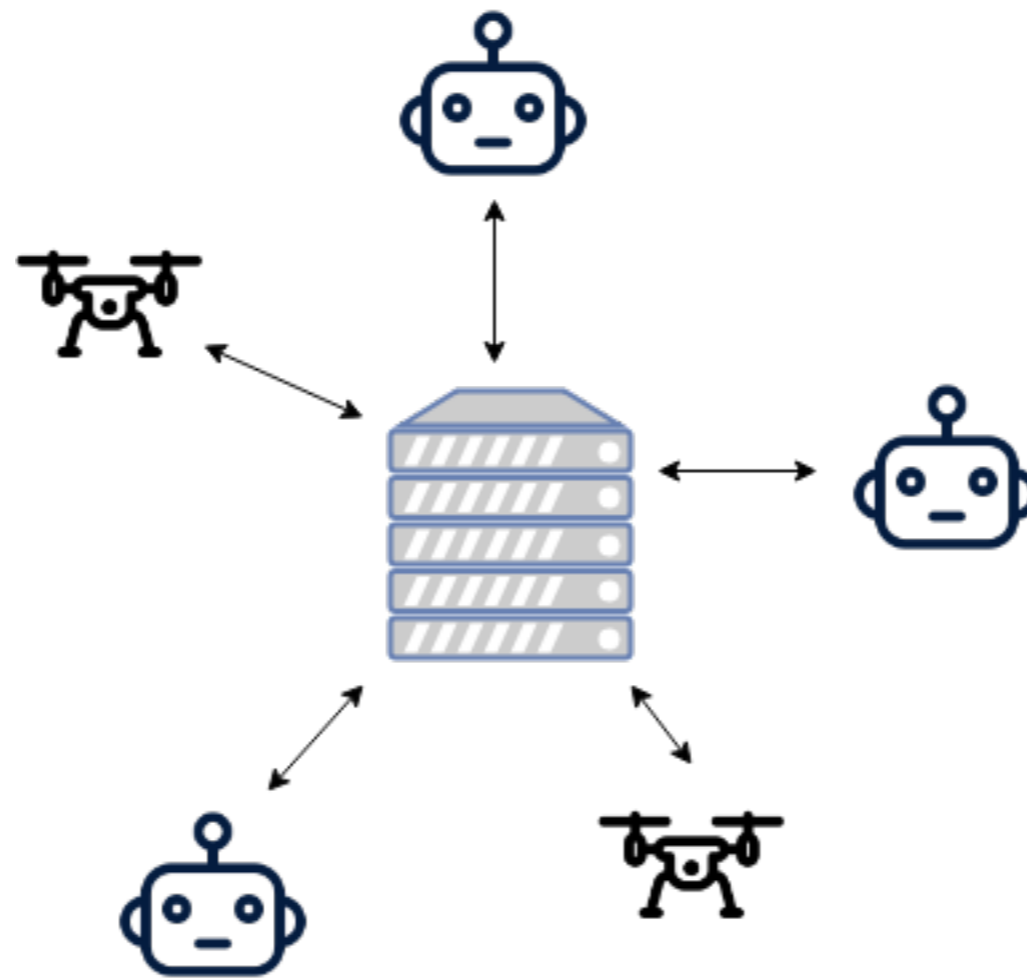


This entire relationship is decomposable!

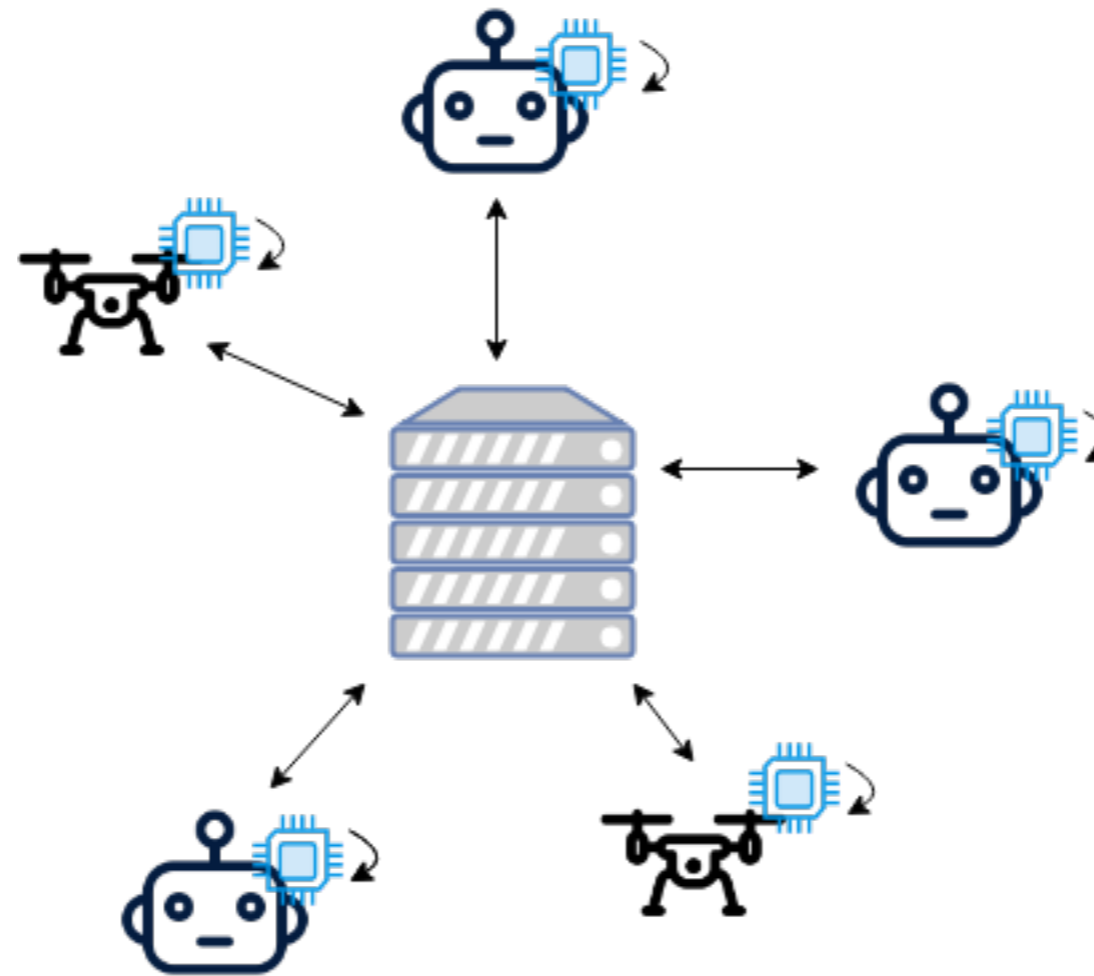
Distribution of computation across hosts



Distribution of computation across hosts



Distribution of computation across hosts



Distribution of computation across hosts

Autonomy vs. computation and **graceful degradation**

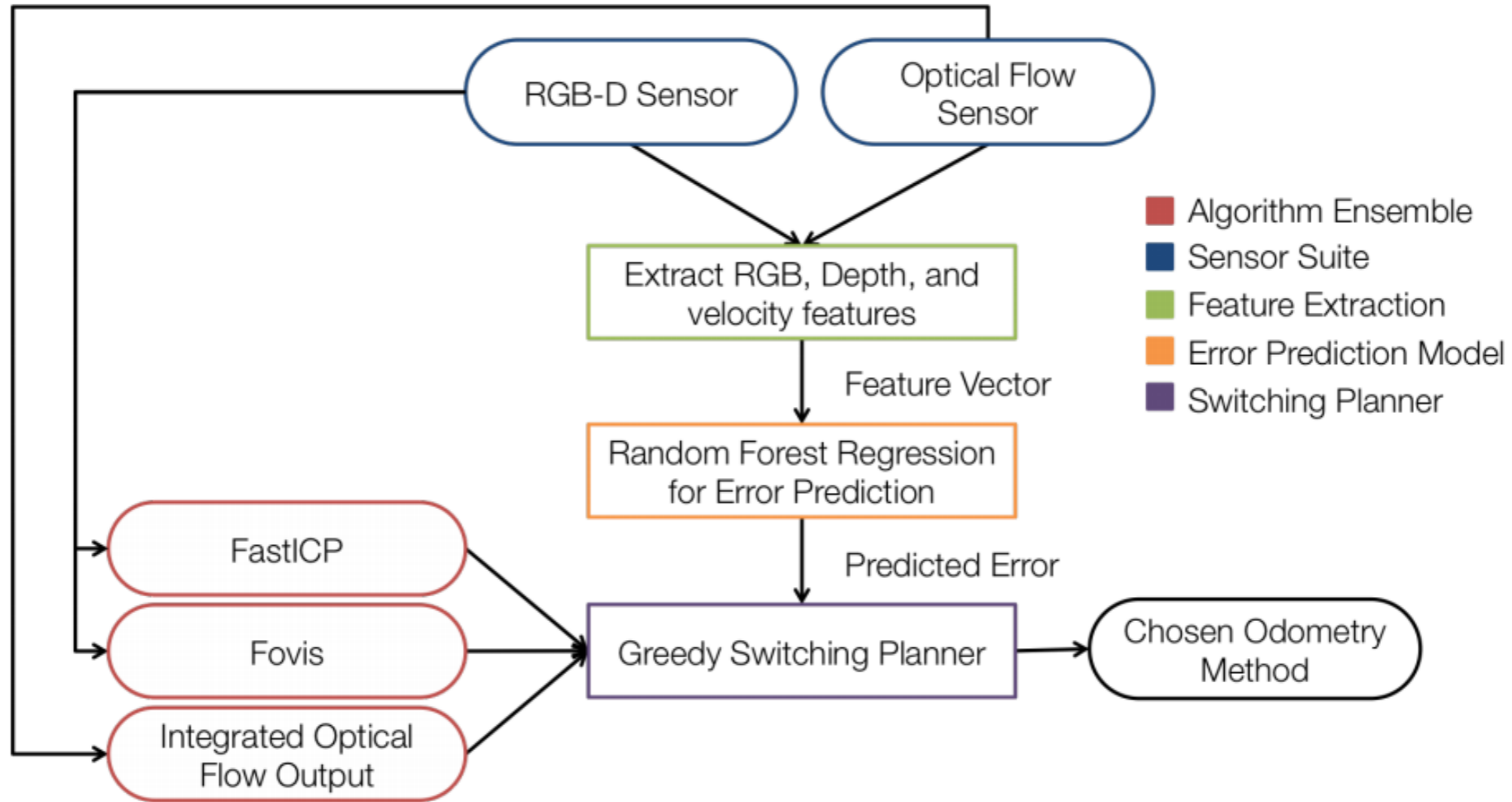


Anomalies in multi-agent systems

For some robot $r \in \text{MARS}$ which is behaving anomalously,

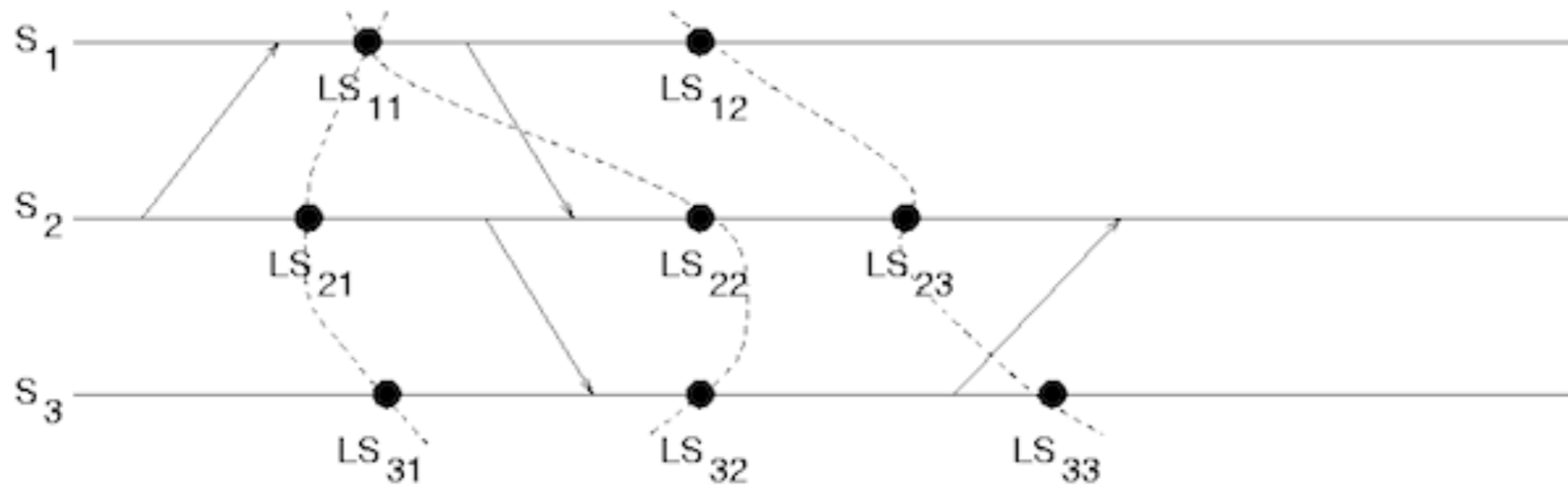
1. How do we take that r out of the system gracefully? (failover)
2. How do other robots in MARS compensate for the lost ability of r ?
3. How can we gracefully re-integrate r into MARS?
4. At what point is the overall task no longer achievable?
5. If the paired a_n does not detect an anomaly in the corresponding r_n , but one does exist, then can the other a_n detect the anomalous behavior? If so, how?

Fixing anomalies on the fly

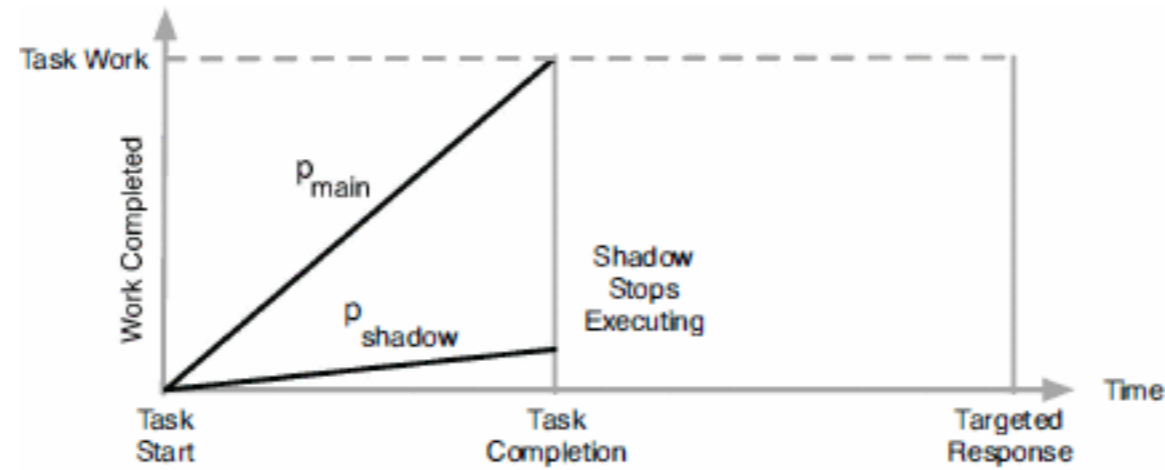


Kristen Holtz, Daniel Maturana, and Sebastian Scherer. "Learning a Context-Dependent Switching Strategy for Robust Visual Odometry."

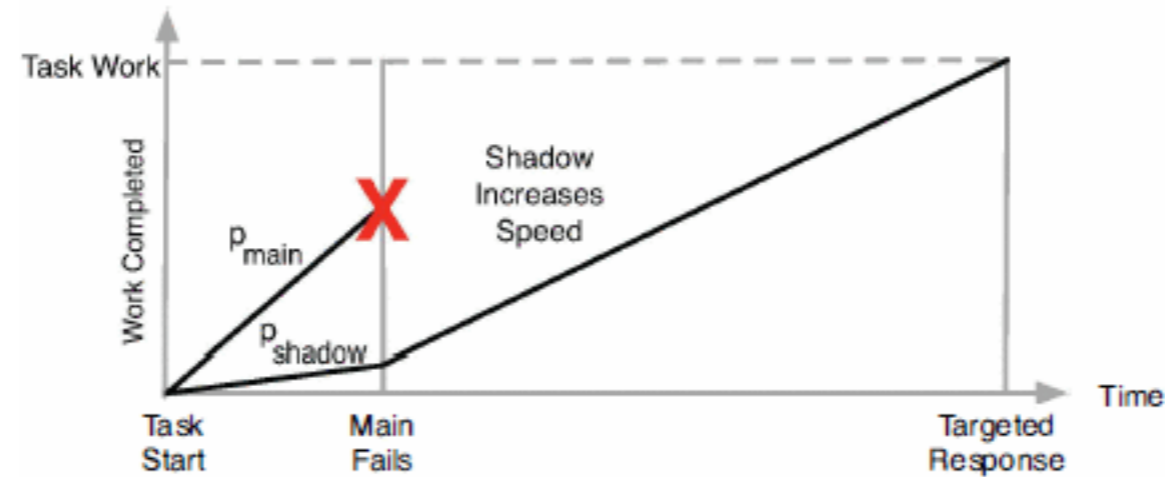
Fixing anomalies on the fly



Fixing anomalies on the fly



(a) Case of no failure



(b) Case of failure

B. Mills, T. Znati, and R. Melhem. "Shadow Computing: An energy-aware fault tolerant computing model."