A Framework for Multi-Agent Fault Reasoning in Swarm Satellite Systems

James Staley, Kerri Lu, Elaine Schaertl Short, Evana Gizzi (evana.gizzi@nasa.gov)
NASA Goddard Space Flight Center, Greenbelt, MD

Abstract

Complex distributed systems are critical to scientific discovery in space. Future missions will take place in increasingly remote planetary environments where human intervention will neither be feasible, nor scalable toward fleet-wide mission control. To this end, autonomous onboard fault mitigation will be necessary. **The unique topology of fleet systems offers opportunities for high-level contextual understanding of faults and coordinated fault mitigation not possible for single agents**. We present a framework that augments single-agent fault mitigation with the context provided by a fleet. Multi-Agent Anomaly Detection (MAAD) operates on time-series sensor data to build a unidimensional distribution against which we can compare individual agents in order to detect faulty sensing hardware.

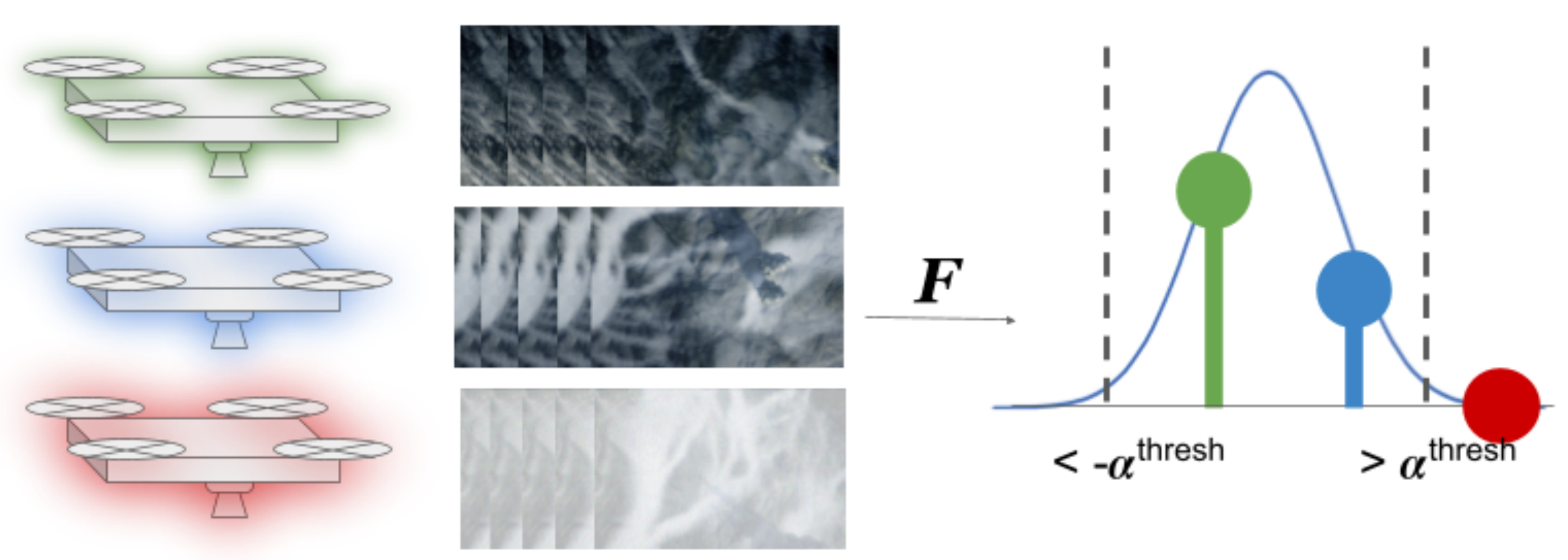

Figure 1: We model windowed drone frames features as a single distribution and flags outliers as anomalous events.

Multi-Agent Anomaly Detection

Given n agents producing S^n sensor frames at each timestep, we model the feature extracted frames $x^1..x^n$ as a single distribution and evaluate the likelihood that each agent falls outside the distribution. The choice of feature extraction function $\boldsymbol{F}$, model construction function $\boldsymbol{G}$, and likelihood estimate function ε will depend on the deployment context. We show autoencoder reconstruction error (AE), mean-square error (MSE), and windowed-absolute gradient for $\boldsymbol{F}$, construct gaussian models $\boldsymbol{G}$, and use z-score thresholding ε to determine outliers.

Algorithm 1 Multi-Agent Anomaly Detection

```
1: $n$ : number of agents
2: $T$ : timesteps in batch
3: $k$ : number of mnemonics in one data frame
4: $d$ : number of features extracted from a data frame
5: $S^n$ : $n$ batches of science frames
6: $\Gamma$ : Set of anomalous agents
7: $\mathcal{F}$: $\mathbb{R}^k \rightarrow \mathbb{R}^d$ feature extraction function
8: $\mathcal{G}$: $\mathbb{R}^{nxdxT} \rightarrow \mathcal{M}$ Model construction function
9: $\mathcal{E}$: $\mathcal{M} \cap \mathbb{R}^{dxT} \rightarrow \mathbb{R}^1$ Likelihood estimation function
10: $\alpha_{th}$ : anomaly threshold value
11: $detecting \leftarrow$ True
12: while $detecting$ do
13:     $S_{\neg\Gamma} \leftarrow S \setminus \Gamma$ Remove known anomalies
14:     $x^i \leftarrow \mathcal{F}(S^i)$ for $i = 1..n$ Extract features
15:     $C \leftarrow x^1...x^n$ Concatenate context
16:     $\mathcal{M} \leftarrow \mathcal{G}(C)$ Create model
17:     $p \leftarrow \mathcal{E}(\mathcal{M}, x^i)$ for $i = 1..n$ Likelihood under model
18:     $detecting \leftarrow$ False
19:     for all $p^i \in p$ do
20:         if $p^i < \alpha_{th}$ then
21:             $\Gamma \leftarrow \Gamma \cup \{i\}$
22:             $detecting \leftarrow$ True Keep looking for anomalies
23:         end if
24:     end for
25: end while
26: return $\Gamma$
```

Simulated Visual Anomalies

Simulation Environment
We controlled a multi-agent drone swarm within the AirSim simulation environment to serve as a proxy satellite fleet. Each drone records a 256x144 color camera image at each timestep. Drones are issued mid level commands that are then carried out by onboard controllers to survey the landscape beneath them.

Fleets of five drones surveyed the surface for ten, 80 second trials producing around 3400 frames of video per drone in total. For both of our anomalous hardware conditions: (1) Gaussian noise, (2) Perlin noise, we apply the anomaly to every frame in a single randomly selected drone's camera feed. Gaussian noise is sampled per frame to simulate electrical noise, while a single randomized Perlin noise map is used for all frames in a trial to simulate accumulated debris partially occluding the camera lens. In addition we simulate randomized rain streaks on all drone cameras to test robustness to environmental sources of noise.

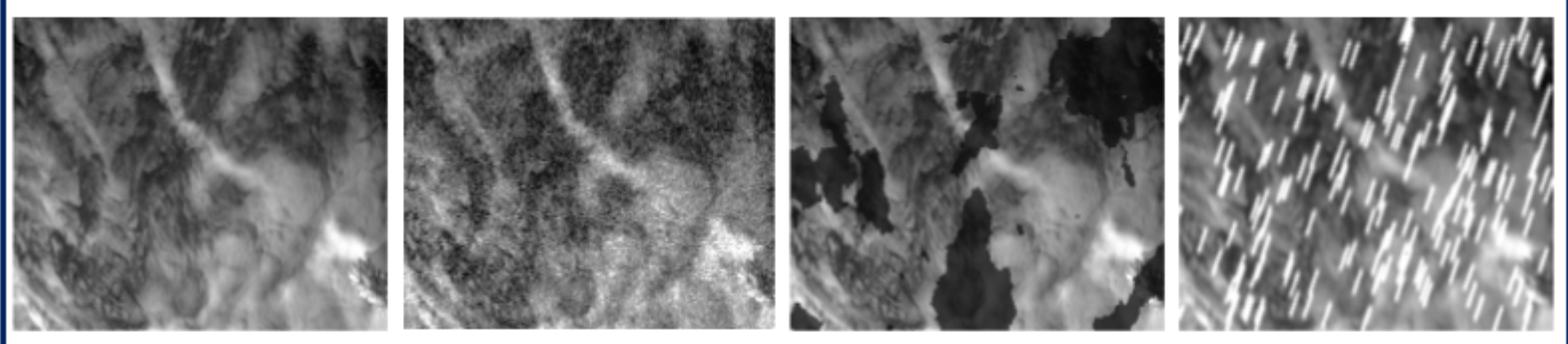

Figure 2: Anomalous images from downward facing drone cameras. Left-to-right: None, Perlin noise, Gaussian Noise, Rain

Simulation Results
We show results for two choices of feature **F**: a non-parametric first-order feature, Mean Squared Error (MSE), and a deep learning feature, the reconstruction error of a convolutional autoencoder (AE). Drones are classified as anomalous or nominal in each two second window.

The separation between each experimental condition and the baseline (none) in figure 3 indicate the anomalies are detectable with a low false positive rate. Drones experiencing no anomaly lie on the black line. In the case of rain (green), all drones experience the environmental anomaly and no anomaly detection signal is raised. Reconstruction error more easily distinguishes between conditions and results in a greater Area Under the Curve (AUC) score (figure 4).

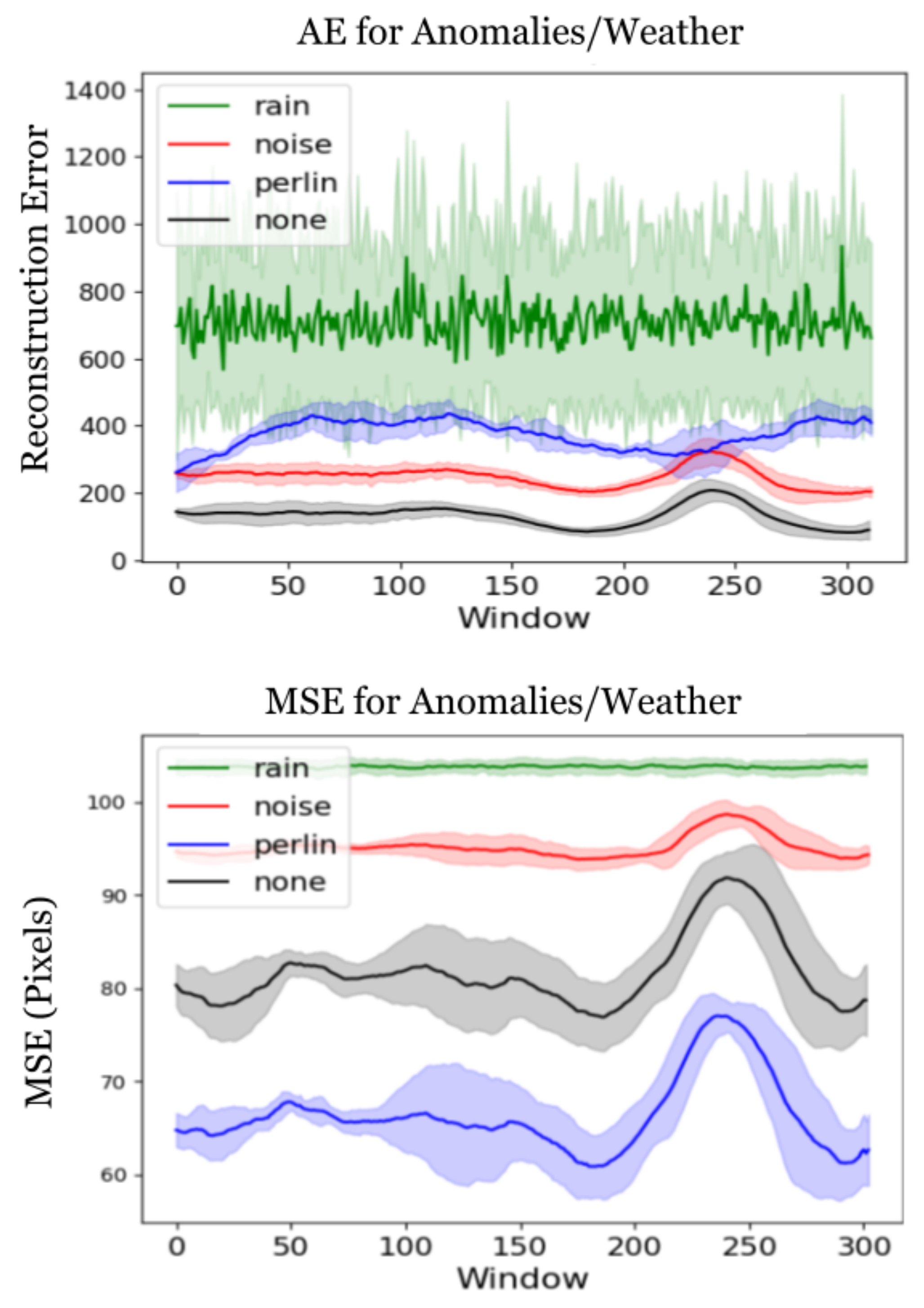

Figure 3: Moving average +/- one standard deviation of Mean Squared Error (MSE) and Convolutional Autoencoder reconstruction error (AE) features for rain, white noise, perlin noise, and control (none) conditions.

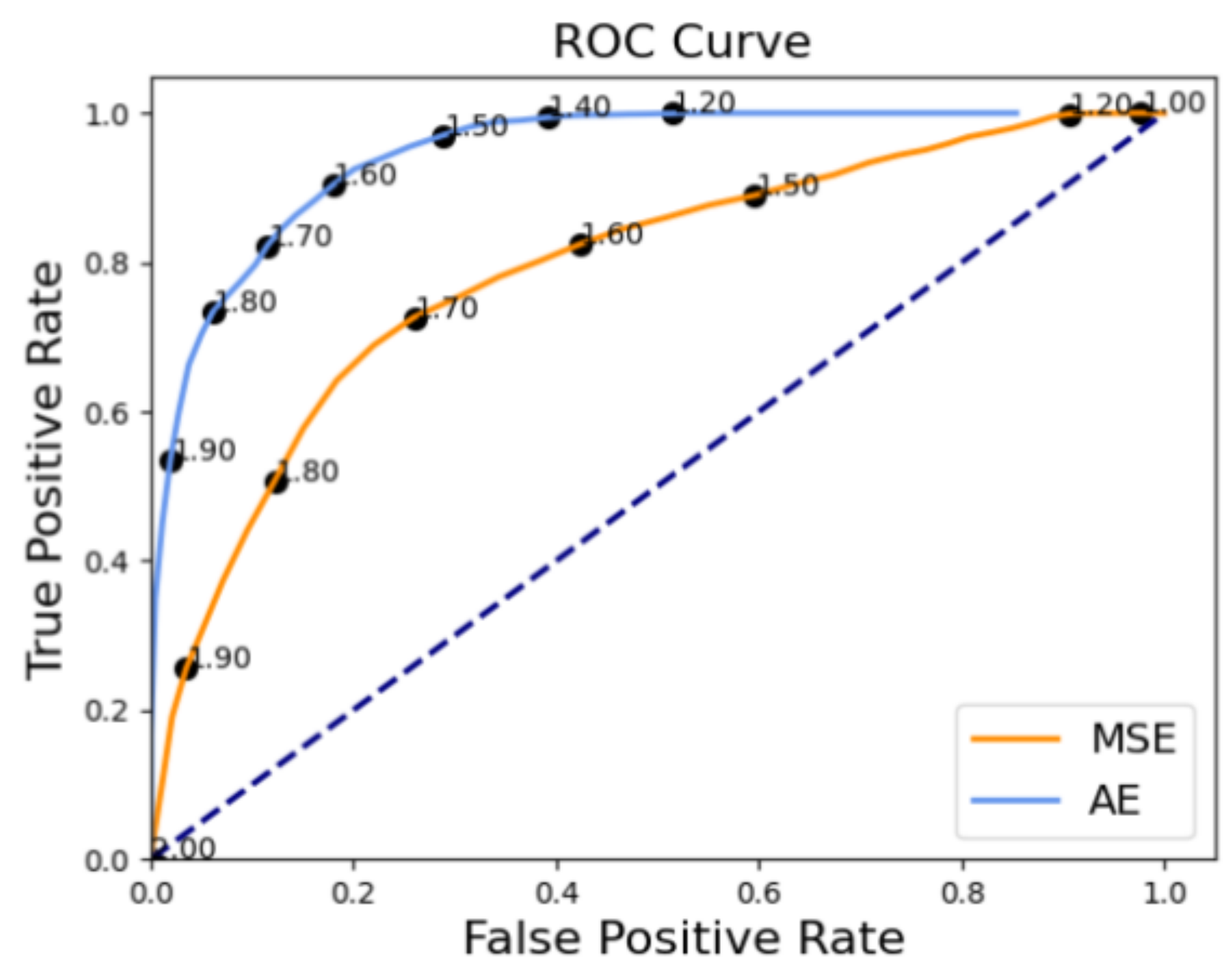

Figure 4: Receiver Operating Characteristic curve for Autoencoder reconstruction error and Mean Squared Error features. The AUC is 0.945 for AE and 0.789 for MSE.

Real World Dataset

Gas Concentration Data
Real-world drones were equipped with scientific instruments with ten methane detecting sensors. Sensors 3 and 10 were non-functioning, while Sensors 1 and 5 display anomalous behavior.

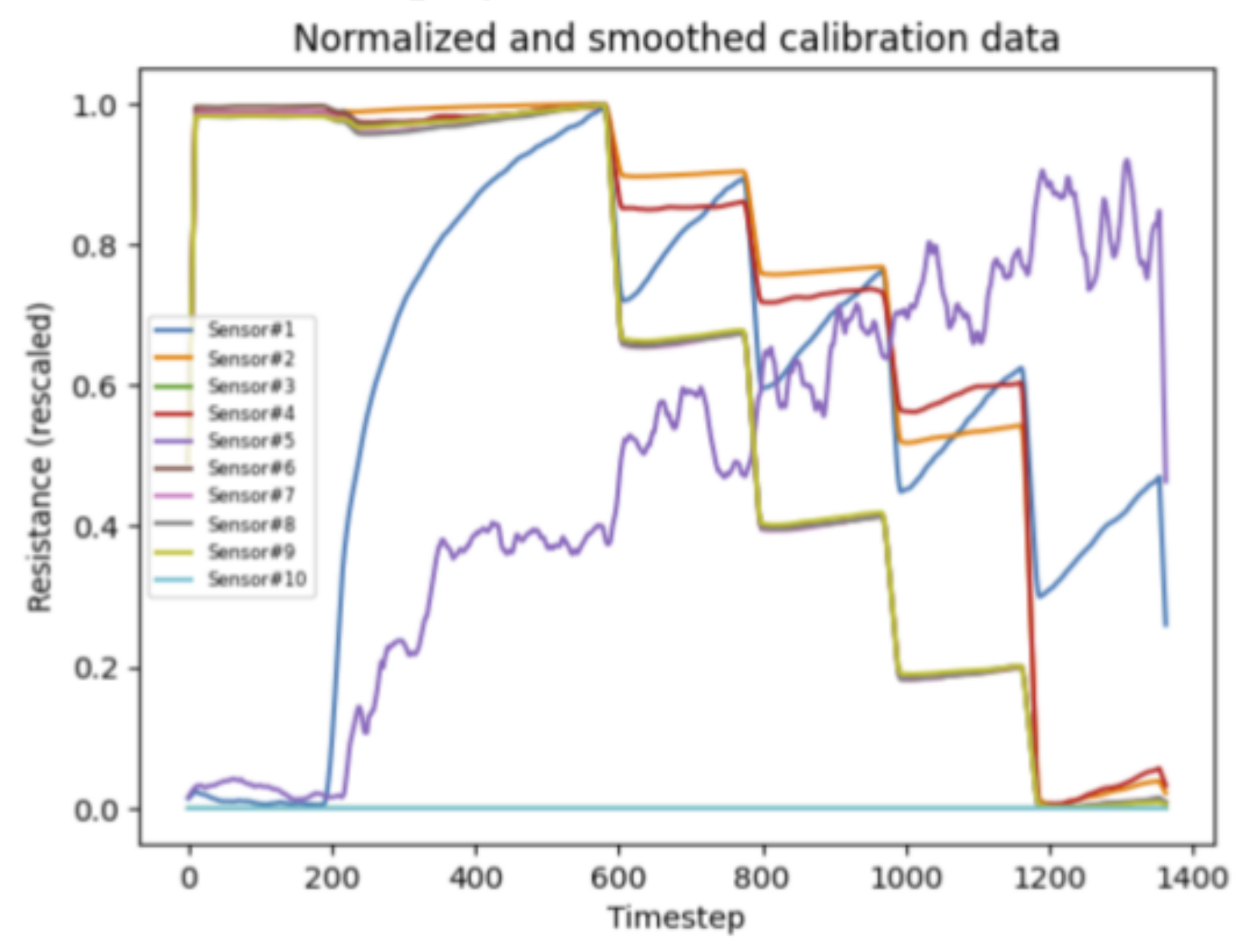

Figure 5: Sensor resistance calibration data from climate monitoring drones. The data is standardized to a 0-1 range and smoothed with a moving average filter to remove noise.

With the functioning sensors, we form a univariate distribution by taking the moving average our feature, the absolute one-step gradient. We find all deviant sensors (1 and 5) by comparing each sensor's z-score under this distribution to a threshold at each step.

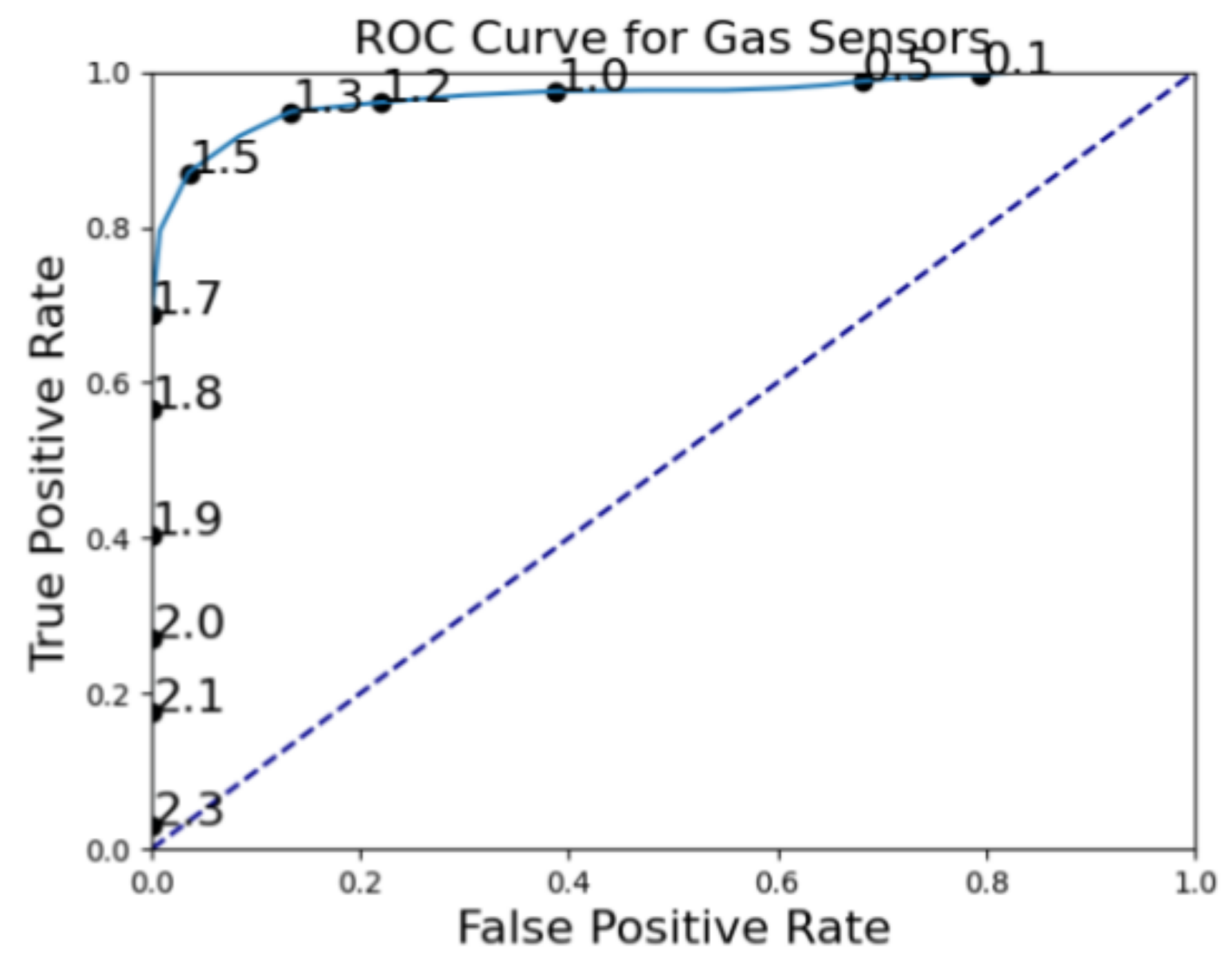

Figure 6: ROC curve for gas sensor anomaly detection using windowed absolute gradient. AUC is 0.97

Conclusion

Our framework performs anomaly/novelty detection by leveraging multiple-agents to contextualize sensor readings, allowing for the differentiation of environmental vs. hardware anomalies. The fleet's task is implicitly considered by assuming similar sensor feed characterizations. Agents that differ from this expectation are flagged as anomalous. **A meta-agent collects the context and looks for anomalies not detectable by a single agent**. This work shows that onboard intelligence can support resilience and novel science discovery in future missions.