Semaphore Dev Server

Model Failure Analysis

Feb 28 – Mar 8, 2026 · 9 days · 106 models · sherlock-dev

Compared against: Feb 8–18, 2026 report

TOTAL RUNS	FAILURES	FAILURE RATE	MISSING DATA	STALE DATA
21,397	617	2.88%	522 (84.6%)	95 (15.4%)

Overall failure rate improved from 6.6% (Feb 8–18) to 2.88% (Feb 28–Mar 8).

Prior Open Items — Status Update

The Feb 8–18 report identified four open questions and proposed actions. This section reviews what the current data reveals about each.

1. Bird Island — Request larger window; check if failing point was the now-point

STATUS: ANSWERED. Yes, the failing point is the now-point in 85% of all LIGHTHOUSE failures (244 of 287 cases). Although there are still some failures that might yet be solved, further analysis below.

2. Magnolia — Enable interpolation on dWnDir (angled) and dWnSpd (normal)

STATUS: ANSWERED. Interpolation on dWnDir (angled) and dWnSpd (normal) has been enabled on magnolia models. However, there are still 3 magnolia failures visible in the current dataset that are all trailing-edge (now-point) failures.

3. NOAATANDC — Possible update window pattern; needs further investigation

STATUS: SIGNIFICANTLY WORSE. NOAATANDC went from 72 failures (4.1% of total) in the prior period to 130 failures (21.1%) in the current period, making it the second-largest failure driver. No fix has been implemented for this source but I talk about investigation steps below.

4. NDFD_JSON — Likely isolated; monitor for recurrence

STATUS: NDFD IS RECURRING. The Feb 8–18 report characterized the Feb 10 NDFD event as “likely an uncommon case” and recommended monitoring without immediate action. The current period shows 4 distinct NDFD stale events in just 9 days (vs. 1 event in 11 days), all with consistent ~11–12 hour time_diff values, indicating NDFD regularly fails to publish its 12-hour update on schedule.

Overview

The Feb 28–Mar 8 window shows a substantially improved baseline compared to the prior period, but the improvement is uneven.

SOURCE	FAILURES	% OF TOTAL	TYPE	CHANGE VS PRIOR PERIOD
LIGHTHOUSE	287	46.5%	missing	Down from 51.7% → now shares burden with NOAATANDC
NOAATANDC	130	21.1%	missing	Up significantly — was 4.1% in prior period
SEMAPHORE	105	17.0%	missing	Cascade from LIGHTHOUSE (magnolia_transform)
NDFD_JSON	95	15.4%	stale	Up significantly — was 5.5% in prior period; multiple events

Daily Failure Rate Trend

DATE	TOTAL RUNS	FAILURES	RATE	DOMINANT SOURCE
Feb 28	2,544	37	1.45%	LIGHTHOUSE + NOAATANDC
Mar 1	2,544	28	1.10%	LIGHTHOUSE (low day)
Mar 2	2,544	36	1.42%	LIGHTHOUSE + NOAATANDC
Mar 3	2,544	57	2.24%	NDFD stale event (29) + scattered
Mar 4	2,544	53	2.08%	LIGHTHOUSE + NOAATANDC
Mar 5	2,542	103	4.05%	LIGHTHOUSE spike + NDFD (6)
Mar 6	2,538	70	2.76%	LIGHTHOUSE + NOAATANDC
Mar 7	1,510	31	2.05%	Partial day
Mar 8	2,087	202	9.68%	NOAATANDC multi-hour gap + NDFD stale (59)

LIGHTHOUSE — Missing Data Analysis

LIGHTHOUSE accounts for **287 failures** (46.5% of all failures), down from 51.7% in the prior period but still the top failure driver. The dominant pattern has shifted: whereas the prior period was dominated by persistent day-long gaps from a permanently missing sensor reading, this period shows a cleaner split between two distinct failure modes.

Is the Missing Point the Most Recent ('Now') Point?

Key Finding: Yes — for 85% of LIGHTHOUSE failures, the missing timestamp is the current hour.

ThermalRefuge — Systematic Every-Run Failures

ThermalRefuge is failing on every single run across the full 9-day period (200/200 failures, 100% failure rate).

DATE	FAILURES	RUNS
Feb 28	24	24
Mar 1	24	24
Mar 2	24	24
Mar 3	24	24
Mar 4	24	24
Mar 5	23	24
Mar 6	24	24
Mar 7	14	~14 (partial day)
Mar 8	19	~19 (partial day)

ThermalRefuge runs at the top of every hour (:00 rather than :22) and fails exclusively on LIGHTHOUSE data. All failures fall in the now-point category, meaning each run is consistently missing the most recent LIGHTHOUSE point. The **complete, uninterrupted failure across all 200 observed runs** strongly suggests a latency issue. We're consistently requesting the point before it is recorded in the LIGHTHOUSE dataset.

> **ACTION REQUIRED: ThermalRefuge needs to be set to run at least 23 minutes after the hour.**

Non-ThermalRefuge LIGHTHOUSE Failures

The remaining 87 LIGHTHOUSE failures (excluding ThermalRefuge) split as follows: 45 are now-point failures on various models, and 42 are the Bird Island/MRE haunted gap failures.

Bird Island + MRE (84): Each dropout produces exactly two failure batches 26 hours apart.

Magnolia (3): Dropout only causes one failure at the right side of the window (now-point). This is because it wasn't a permanently lost point, just a latency on the surge value, so we might want to run these even later than *:22, although this only happened once in the time frame so might not be a large issue.

Why Interpolation and the Larger Window Are Not Catching These Failures

The full dataset (Feb 28–Mar 8) yields 287 LIGHTHOUSE failures splitting cleanly into 245 now-point failures (6-hour lag) and 42 interior haunted gap failures (32-hour lag). Both categories persist despite interpolation being enabled and the larger-window fix being proposed.

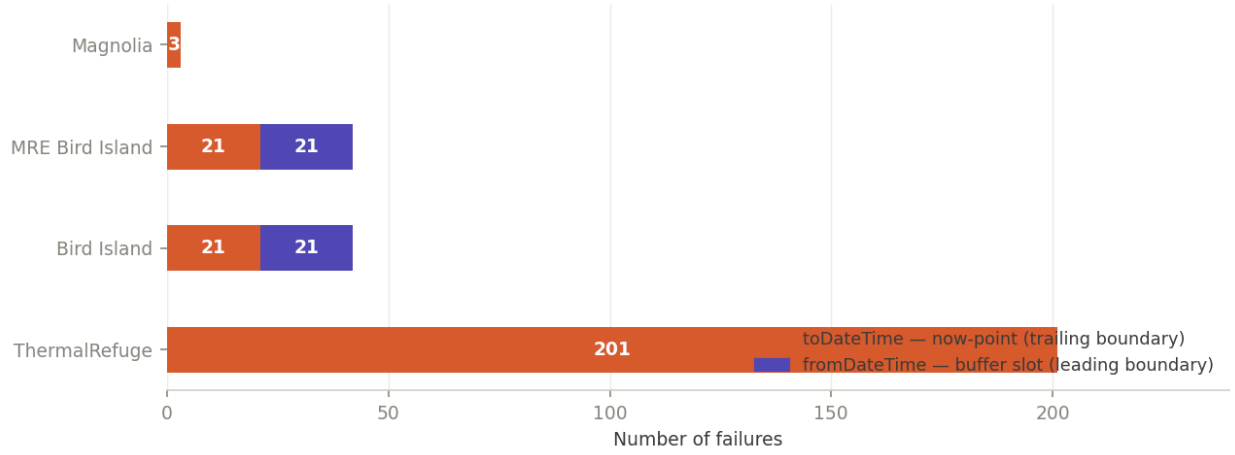

Figure 1: LIGHTHOUSE failures by model family and window position (Feb 28–Mar 8, 2026)

Category 1: Now-point failures (245 cases)

Every now-point failure shares the same structural cause: because it is the last point in the window, interpolation has no right anchor and cannot fill it.

Four concrete examples from the data:

Example A — ThermalRefuge (02/28/26 00:00 CST):

```
Run: 00:00:32 CST (06:00 UTC)  |  toDateTime: 2026-02-28 00:00+00:00  |  Missing:
2026-02-28 00:00+00:00 == toDateTime
```

Example B — Bird-Island_Water-Temperature_3hr (03/05/26 19:22 CST):

```
Run: 19:22 CST (2026-03-06 01:22 UTC)  |  toDateTime: 2026-03-05 19:00+00:00  |
Missing: 2026-03-05 19:00+00:00 == toDateTime
```

Extending toDateTime to 20:00+00:00 would require 2026-03-05 20:00 UTC to exist in LIGHTHOUSE. At 01:22 UTC that slot is not yet ingested. No right anchor materializes; the larger-window fix does not resolve this case.

Example C — magnolia_12 (03/04/26 18:22 CST, dWnDir interpolation IS enabled):

```
Run: 18:22 CST (2026-03-05 00:22 UTC)  |  toDateTime: 2026-03-04 18:00+00:00  |
Missing: 2026-03-04 18:00+00:00 == toDateTime
```

Interpolation on dWnDir/dWnSpd is enabled. It still fails because the missing point is the trailing edge — no right anchor exists regardless of interpolation config. LIGHTHOUSE CSV confirmed (station 057 Port O'Connor): dWnSpd and dWnDir are still NA at exactly 2026-03-04 18:00 UTC; readings at 17:54 and 18:06 UTC are valid. This is a permanent single-slot sensor transmission failure, not latency. Station 057 shows 6 scattered NA slots across Mar 4 (05:18, 15:48, 16:12, 18:00, 20:48, 23:48) consistent with intermittent transmission failures. The T+24h run (03/05/26 12:22 CST) confirmed successful via log: the dropout was interior to the window and interpolated as expected.

Category 2: Interior haunted gap failures (42 cases) — limit_area='inside' refuses fromDateTime boundary; DateRangeValidation fails on unused buffer slot

All 84 Bird Island and MRE LIGHTHOUSE failures in this dataset trace to a single dropout at 2026-03-05 19:00 UTC (South Bird Island). That one dropout produced exactly two failure batches — no more, no fewer. Understanding why requires tracing the full lifecycle of a dropout through the sliding window.

The dropout lifecycle (range [0, -26], interval 1hr):

```
T+0h  (03/05/26 19:20 CST):  dropout == toDateTime     → now-point, FAILS    (42
models)
T+1h  to T+25h:  dropout is INTERIOR       → interpolated, SUCCEEDS (~1,050 runs)
```

```
T+26h (03/06/26 21:20 CST):  dropout == fromDateTime    → buffer slot, FAILS  (42
models)
T+27h onwards :  dropout outside window    → gone, SUCCEEDS
```

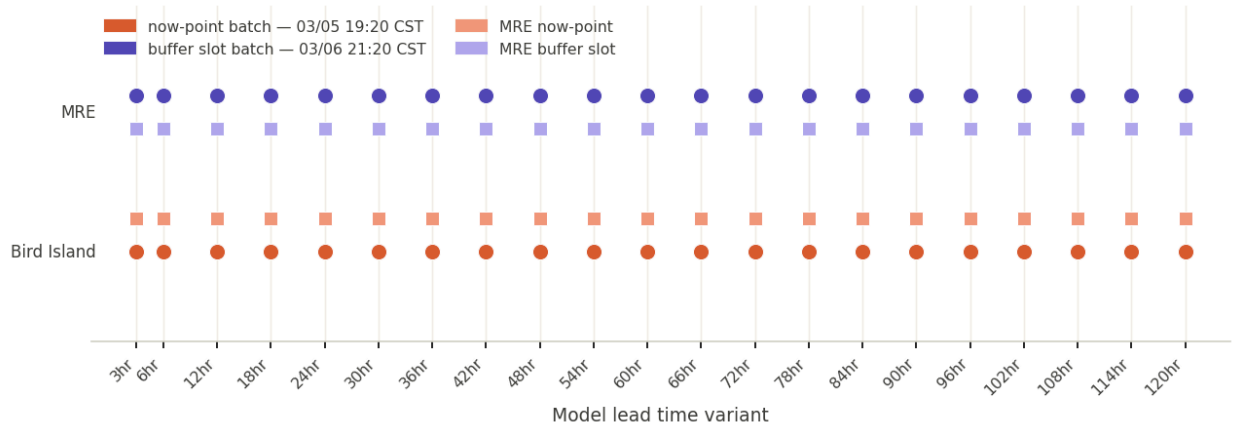

Figure 2: All Bird Island and MRE Bird Island models — every lead-time variant failed in both the now-point batch (03/05 19:20 CST) and buffer-slot batch (03/06 21:20 CST), confirming a single sensor dropout swept the entire fleet

Example D — Bird-Island_Water-Temperature_3hr (03/06/26 21:21 CST):

```
Run: 21:21 CST (2026-03-07 03:21 UTC)  |  toDateTime: 2026-03-06 21:00+00:00
Missing: 2026-03-05 19:00+00:00  (26hr before window end = INTERIOR)
Left anchor: 2026-03-05 18:00 UTC (present)   Right anchor: 2026-03-05 20:00 UTC
(present)
```

The deeper issue is that the model doesn't actually need the −26 point. The **vectorOrder** only indexes **[0, 25]** — 26 data points, corresponding to **range [0, -25]**. The −26 slot is a pure buffer, requested only to give **limit_area='inside'** a left anchor so it can fill a gap at −25. But **DateRangeValidation** validates the *entire requested window* including the buffer slot.

DataGatherer validates all 27 slots (range [0, -26]) but the model only ever reads 26 slots (vectorOrder [0, 25]). DateRangeValidation doesn't know slot -26 is a buffer it's allowed to ignore.

Fix Idea: in __validate_series (or just before calling it), derive the validation start time from vectorOrder instead of timeDescription.fromDateTime.

```
#Find this series' entry in vectorOrder
vector_entry = next(
    (v for v in dependentSeries_list if v.outKey == key),
    None
)

if vector_entry is not None:
    max_index = vector_entry.indexes[1]  # e.g. 25
    validation_from = timeDescription.toDateTime - timedelta(
```

```
        seconds=max_index *
timeDescription.interval.total_seconds()
    )
else:
    validation_from = timeDescription.fromDateTime  # fallback, no
change

# Pass validation_from to DateRangeValidation instead of
timeDescription.fromDateTime
```

Nothing new in the dspec. No extrapolation. vectorOrder was already there — we're just using it for something it logically should already be driving. The buffer slot exists for interpolation, validation checks only what the model reads. Those two things finally agree.

The only code that changes is DataGatherer.__request_dependent_data — it needs to pass the vectorOrder-derived start time down to __validate_series. DateRangeValidation itself needs a way to accept a custom fromDateTime override, but that's a small constructor argument, not a new concept.

SEMAPHORE Source — Cascade Failures

The 105 SEMAPHORE-source failures are not independent. They represent the **magnolia_transform** models failing because their upstream inputs (magnolia_12/24/48 outputs) are absent when LIGHTHOUSE fails. The dependency chain is unchanged from the prior period:

MODEL	FAILURES
magnolia_transform_12	35
magnolia_transform_24	35
magnolia_transform_48	35

The uniform count of 35 failures per transform model indicates these downstream failures track 1:1 with the upstream magnolia failures. No action is needed specific to the SEMAPHORE source — fixing the LIGHTHOUSE issues above will resolve these.

NOAATANDC — Missing Data Analysis

NOAATANDC is the most significant **new development** in this period. It jumped from 72 failures (4.1% of total) in the prior report to **130 failures (21.1% of total)**.

Failure Volume Comparison

PERIOD	TOTAL NOAA FAILURES	% OF ALL FAILURES	WORST DAY
Feb 8–18 (prior)	72	4.1%	Feb 10: 21 failures
Feb 28–Mar 8 (current)	130	21.1%	Mar 8: 67 failures

A lot of analysis of this issue had to be done so a separate document was created.

NDFD_JSON — Stale Data Events

NDFD_JSON stale failures increased from **62 failures across 1 event** in the prior period to **95 failures across 4 events** in this period. All NDFD failures are stale-data type (not missing), with a characteristic time_diff of approximately 11–12 hours — NDFD had not updated within the staleness threshold.

DATE	FAILURES	TIME_DIFF (APPROX)	NOTES
Feb 28	1	11h 43m	Isolated single-model failure
Mar 3	29	~11–12h	Coordinated multi-model event; 29 models failed simultaneously
Mar 5	6	~11–12h	Smaller event; partial recovery
Mar 8	59	~11–12h	Largest event; coincided with NOAATANDC outage

The prior report described the Feb 10 NDFD event as **"likely isolated"** and recommended monitoring. This period shows NDFD is **not isolated — it is recurring**. Three distinct events occurred in 9 days (vs. 1 in 11 days), and the largest event (Mar 8, 59 failures) occurred on the same day as the major NOAATANDC outage, compounding the worst day in the dataset.

The consistent ~11–12 hour time_diff across all events suggests NDFD is regularly failing to publish its 12-hour update cycle on schedule. This is distinct from a rare isolated delay — it is a recurring pattern that warrants the same staleness offset treatment previously applied to NDFD_JSON.

> **RECOMMENDATION: Revisit the NDFD_JSON staleness threshold or offset. Given three events in 9 days with consistent ~12hr time_diff, a larger staleness offset (e.g., 13–14 hours) may suppress false-positive failures during normal NDFD update delays, consistent with how offsets were previously calibrated for other sources.**
